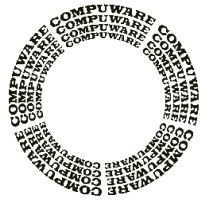
Abend-AID XLS
User Reference Guide

Release 9.4



COMPUWARE®

Please direct questions about Abend-AID XLS
or comments on this document to:

Abend-AID XLS Technical Support
Compuware Corporation
31440 Northwestern Highway
Farmington Hills, MI 48334-2564
1-800-538-7822

Outside the USA and Canada, please contact
your local Compuware office or agent.

Doc. CWAAUM9E
July 26, 2002

Contents

Figures

Tables

Summary of Changes

This section summarizes the enhancements to the most recent releases of Abend-AID XLS.

Release 9.4 Changes

Abend-AID XLS release 9.4 introduces significant improvements in the areas of installation, customization, and usability. Note that Abend-AID XLS 9.4 requires Compuware Shared Services Release 7.9 or more current.

- **Installation Process**
  - Abend-AID XLS will be installed using IBM's System Modification Program Extended (SMP/E) facility.
  - A new panel-driven, installation dialog will gather the information that is specific to each site and generate the JCL and customization modules to install the base product and all licensed options.

The installation process requires TSO/ISPF.

- **Customization**
  - Site-specific customization will be completed through the installation dialog.
  - The CWGLOBAL load module will be updated and created using the installation dialog.
  - A new customization load module, CWROUTE, will be created using the installation dialog to store the names of site default Shared Directories as well as information for routing Abend-AID XLS reports.
  - Reports can be routed to specific report databases at abend time based on specific criteria unique to the abending job. This can be used to replace the use of CWJOBTAB and CWEXIT02 used in previous releases to route reports to a specific report file.
  - AAXTERM1 and AAXTERM3, user-coded exits available in previous releases of Abend-AID XLS, are now standard features. The report dataset that the Abend-AID XLS report is written to and the report number will be included in the job output.
  - A source-merged report can be written to SYSOUT using the report routing options.
- **Starting the SVC51 Interface**
  - The JCL used to start the SVC51 Interface has changed. A //CWROUTE DD card is required in the JCL and points to a dataset containing the CWROUTE load module.
- **Report Format**
  - The formats of the Header, Diagnostic, NSI, Call Trace, and Register sections have been modified to provide information in a concise format. When using the Compuware/VF, users, by default, go directly to the Error Analysis Section of the report.
- **Report and Source Listing Datasets**

  - All Abend-AID XLS reports will be written to report datasets that can be viewed using the Compuware Viewing Facility.
  - Abend-AID XLS will use Shared Directories and attached Databases to access reports. The capacity of the shared directory can be increased without reallocating files.
  - Source Listing Databases attached to Shared Directories will be used to store Source Listings. These Source Listing Databases can be shared by CICS Abend-AID/FX, XPEDITER/TSO, and XPEDITER/CICS).

- **VisualAge for PL/I and Enterprise PL/I Extended Language Support**
  - This release includes support for source merge of VisualAge PL/I and Enterprise PL/I programs.
- **Accessing the Compuware/VF from SDSF**
  - By executing a CLIST in SDSF, the Compuware/VF will be invoked, and the user will be taken directly to the report that was created for that job.
- **Support for Earlier Releases**
  - Support for Abend-AID XLS 9.2 will be discontinued on March 1, 2003.
  - Abend-AID XLS 9.4 will be the last version of the product to support the following IBM software:
    - MVS/ESA
    - OS/390 V1 and associated versions of Language Environment including LE for MVS and VM 1.5
    - DB2 3.1
    - IMS 4.1 and 4.9

Release 9.3 Changes

- **Language Environment (LE) Changes:**

  All LE-specific information related to an error is now provided in a new LE Section of the Abend-AID XLS report. This centralized repository of information, coupled with Abend-AID XLS's powerful, new diagnostic algorithms, enables programmers to resolve errors more quickly than normally possible using IBM's CEEDUMP, IPCS, or a department LE expert.

  In particular, Abend-AID XLS displays the following for the LE User Heap analysis:

  - User Heap run-time parameters and variables including initial Heap size, increment size, number of segments, pointer to first segment, and pointer to last segment.
  - For each segment in the User Heap:
    - Analysis summary of each segment in the User Heap:

      > number of element
      > size of allocated storage
      > number of free elements
      > size of storage still available in segment
      > number of element allocation errors

    - A display, by field, of each header for each segment.
    - A display of the allocation map for the User Heap. An entry is displayed for each element in each segment with its associated data address, data length and status (allocated or free).

- Validation for each element in a segment for allocation errors. If an allocation error is detected, the element is identified and the following is provided:

  > Storage around the element in error.
  > Storage that the elements in error are pointing to.
  > Storage of the previous (last good) element, which may be suspect in overlay errors.

- **RRSAF Attach support for DB2:** This release includes support for the Recoverable Resource Manager Services Attachment Facility (RRSAF) introduced by IBM in DB2 5.1.

- **Abend-AID E-Business Edition:** This release supports the new Compuware Abend-AID E-Business Edition product that provides enhanced debugging capabilities for MQSeries batch and IMS applications.

  Note: The new name for Abend-AID XLS E-Business Edition is now Abend-AID for WebSphere MQ. Abend-AID XLS 9.4 supports Abend-AID for WebSphere MQ 2.1.

  Call your sales representative at 1-800-521-9353 for more information.

Introduction

This manual describes how to use Compuware's Abend-AID XLS. The manual consists of the following chapters:

Chapter 1, "Product Overview": Summarizes Abend-AID XLS's facilities and operation.

Chapter 2, "Getting Started": Introduces the Abend-AID XLS report.

Chapter 3, "Diagnosing a COBOL Data-Related Error": Shows you typical diagnostic steps with an Abend-AID XLS report.

Chapter 4, "Diagnosing a PL/I Data-Related Error": Shows you typical diagnostic steps with an Abend-AID XLS report.

Chapter 5, "Diagnosing an Assembler Data-Related Error": Shows you typical diagnostic steps with an Abend-AID XLS report.

Chapter 6, "Diagnosing VS FORTRAN Errors": Summarizes diagnostic considerations.

Chapter 7, "Understanding Database Diagnostics": Introduces you to Abend-AID XLS diagnostics for DB2, IMS, and IDMS.

Chapter 8, "Using SNAP-AID": Explains how to use the SNAP-AID function.

Chapter 9, "Using Abend-AID DD Statements": Lists and summarizes the purposes of Abend-AID XLS's DD statements.

Chapter 10, "Understanding Language Processors": Everything you need to know to get the most from a Compuware language processor.

Chapter 11, "Using Abend-AID in Language Environment": Answers typical questions about diagnostics and operation in Language Environment.

Chapter 12, "Report Samples": Describes the Abend-AID XLS report sections.

Chapter 13, "Understanding the Compuware/VF Interface": Explains screen fields.

Chapter 14, "Printing & Utilities": Summarizes the management facilities available for both installation and operation.

Appendix A, "Testing Samples": Lists sample faulted programs provided with Abend-AID XLS for testing and training.

Glossary: Defines Abend-AID XLS facilities, features, and associated terms.

Intended Audience

This manual is intended for use by application programmers.

Publications

Abend-AID XLS user documentation is now shipped in three online formats on CD-ROM with Abend-AID XLS: Adobe Acrobat PDF, HTML, and BookManager.

- Acrobat enables unlimited user printing from a Web site or workstation

- HTML enables unlimited user viewing from a Web site
- BookManager ensures precise, fast search capabilities from a network server or workstation.

The *Abend-AID XLS Installation and Customization Guide* and the *Enterprise Common Components Installation and Customization Guide* are shipped in paper format. Complete documentation sets should be printed as needed from the Acrobat files supplied on the documentation CD.

The following documents comprise the documentation set that is supplied with Abend-AID XLS:

- *Abend-AID XLS Installation and Customization Guide:* Provides instructions for installing Abend-AID XLS and information about setting it up to fit the development and system configurations at your site.
- *Abend-AID XLS User/Reference Guide:* Provides guidelines and instructions for using Abend-AID XLS: basic language support, extended language support (XLS), SNAP-AID, Abend-AID for DB2, Abend-AID for IMS, Abend-AID for IDMS, and Abend-AID XLS DD statements.
- *Abend-AID XLS SMF Cost Analysis Tool User's Guide:* Describes the SMF Cost Analysis Tool, reports, usage, customization, and installation.
- *Enterprise Common Components Installation and Customization Guide*: Provides installation and maintenance instructions for Enterprise Common Components (ECC): License Management System, Compuware Shared Services, Distributed Viewing Facility, or Host Communications Interface (HCI), as well as all components of extended language support (XLS).
- *Enterprise Common Components Messages and Codes:* Describes error messages generated while installing ECC or while using its components: License Management System, Compuware Shared Services, Distributed Viewing Facility, or Host Communications Interface.
- *Compuware Shared Services User/Reference Guide*: Provides operation and reference information about the components of Compuware Shared Services, which comprise Extended Language Support: Compuware/VF and the Compuware language processor, Compuware's DDIO report and source listing datasets, and the batch file utilities (CWDDSUTL, CWFXSDUT, and CWDDLPUT).
- *License Management System User/Reference Guide:* Provides operation and reference information about the Compuware licensing system that lets you establish, maintain, diagnose, and upgrade your access to the Compuware products licensed at your site.

Online Help

You will find information about using Abend-AID XLS components available through the Abend-AID XLS Primary Menu and the F1 key.

FrontLine Support Web Site

You can access online technical support for Compuware products via our FrontLine support web site. You can read or download documentation, frequently asked questions, and product fixes, or directly e-mail Compuware with questions or comments. To access FrontLine, you must first register and obtain a password at **http://frontline.compuware.com**.

Online Documentation

Documentation for this product is provided on CD-ROM in several electronic formats. PDF files can be viewed with the free Adobe Acrobat Reader, available at

http://www.adobe.com. HTML files can be viewed with any standard web browser. BookManager softcopy files can be viewed with any version of IBM BookManager READ or the IBM Softcopy Reader. To learn more about BookManager or download the free Softcopy Reader, go to **http://booksrv2.raleigh.ibm.com**.

World Wide Web

Compuware's site on the World Wide Web provides information about Compuware and its products. The address is **http://www.compuware.com**.

IBM Documentation

More information related to Abend-AID XLS can be found in the following IBM publications:

- *z/OS Language Environment Customization Guide*: Discusses Language Environment for MVS.
- *z/OS Initialization and Tuning Reference:* Contains information about APF authorization and the use of (E)LPA.
- *z/OS MVS Programming: Authorized Assembler Services Guide:* Contains information about APF authorization.

Technical Support

If you need information about Abend-AID XLS's operating or performance capabilities, or assistance in troubleshooting, first refer to the Abend-AID XLS documentation set. Complete online documentation is provided on CD-ROM with each release of Abend-AID XLS, and is also accessible for online reading or downloading from the Compuware FrontLine Support Web site.

If you call Abend-AID XLS Technical Support for troubleshooting assistance, provide the following information applicable to your problem so that we can help you as quickly as possible.

- The client number displayed in the header of the Abend-AID XLS report.
- The Abend-AID XLS release number displayed in the header of the Abend-AID XLS report.
- The operating system and release in use displayed in the header of the Abend-AID XLS report to help determine operating system dependencies.
- All printed output from the job in question, including console messages, JCL stream, Abend-AID XLS output, and the standard dump (SYSUDUMP, SYSABEND, or SYSMDUMP).
- Output from jobs that were used to install or customize Abend-AID XLS. This output includes the report produced by the CWVERIFY utility. $11VERIF is the Abend-AID XLS installation library member used to execute CWVERIFY.

Abend-AID XLS **Technical Support**
Compuware Corporation
31440 Northwestern Highway
Farmington Hills, MI 48334-2564

1-800-538-7822

Chapter 1.
Product Overview

Operating System Environment — (Consult FrontLine for Updates)

(Consult http://frontline.compuware.com for updates.)

Abend-AID XLS identifies and describes application program failures in the following MVS-based languages and databases:

Programming Language Support

- Language Environment:
  - Language Environment associated with current releases of z/OS and OS/390
  - Language Environment for MVS & VM 1.5*
- COBOL:
  - Enterprise COBOL 3.1
  - COBOL for OS/390 & VM 2.1, 2.2*
  - COBOL for MVS & VM 1.2*
  - AD/Cycle COBOL/370 1.1, 1.0
  - VS COBOL II 1.4, 1.3.2, 1.3.1
  - OS/VS COBOL 2.4
- PL/I:
  - Enterprise PL/I 3.1
  - VisualAge for PL/I 2.2
  - PL/I for MVS & VM 1.1.1*
  - AD/Cycle PL/I 1.1
  - PL/I 2.3
- Assembler:
  - High Level Assembler (HLASM) 1.4, 1.3, 1.2, 1.1
  - Assembler H
- C:
  - C/C++ for MVS & VM*
  - AD/Cycle C/370
  - SAS/C
- FORTRAN
  - VS FORTRAN 2.6, 2.5

Database Support

- DB2 7.1, 6.1, 5.1, 4.1, 3.1
- DB2 RRSAF Attachment
- IMS 8.1, 7.1, 6.1, 5.1, 4.9, 4.1
- IDMS 15.0, 14.0, 12.01

Innovative Access Method (IAM) Support from Innovation Data Processing:

- IAM 7.0, 6.4, 6.3, 6.2, 6.1

*Compuware supports the OS/390 and MVS portion of this product.

Facilities

Abend-AID XLS intercepts and analyzes application program failures to provide comprehensive fault diagnostics, reacting specifically to the type of error, the language of the failing program, and the types of files or databases in use. A concise, easy-to-read report, available online or in printed form, presents the most important information first and includes only the data necessary to solve the problem.

Abend-AID XLS provides diagnostics for DB2, IMS, and IDMS databases as well as COBOL, PL/I, Assembler, C, and VS FORTRAN languages. This support includes those versions of COBOL and PL/I that can be run with Language Environment.

For COBOL, PL/I, and Assembler errors, Abend-AID XLS merges actual source code and diagnostic data into a single report. Online, menu-driven access enables quick and accurate problem resolution. The specialized diagnostics for DB2, IMS, and IDMS environments enable programmers to resolve errors independently of the DBA staff.

Abend-AID XLS Report

The Abend-AID XLS report consists of these sections, which may be accessed individually online by menu selection:

- **Header:** Includes environmental information about the program in error, the operating system release level, the date and time when the error occurred, job name, step name, and JES ID.
- **Error Analysis section:** Includes a comprehensive explanation of the error condition, specific information about the field(s) contained in the statement in error, and suggested courses of action whenever possible.
- **Error Location section:** Provides information for locating the statement in error. When using Extended Language Support, identifies the exact location of the error. The compile date and length of the program in error are also identified.
- **Call Trace Summary section:** Provides the Call Trace Summary and the Application Program Attributes. The Call Trace Summary includes a trace of called/linked programs on the save area chain and indicates the program in error. The Application Program Attributes include the program name, language, compile date, and length for each application program.
- **Supporting Environmental Data section:** Provides supporting environmental data. This information includes the abending Program Status Word (PSW), program PSW, entry point and load point addresses, and the contents of the registers at the time of the error.
- **Program Storage section:** Formats program storage for application programs on the calling chain. Program storage is formatted based on the programming language.
- **Program Listing section:** Available only with COBOL, PL/I, or Assembler extended language support. Provides the procedure divisions for COBOL programs on the calling chain and source listings for PL/I or Assembler programs on the calling chain with the current statement indicated.
- **File section:** Gives a status of each open file along with formatted current and previous records.

- **Specialized Database Support section:** Provides database-specific support for DB2, IMS, and IDMS if the Abend-AID database support facility for that environment is installed on the system.
- **SORT section:** Includes the current COBOL internal sort information. Available only for COBOL programs with internal sort.
- **COBOL Data Locator section:** Lists occurrences and identifies locations of all or specified data or date strings and formats. Available only during online viewing of COBOL extended language support.
- **Abend-AID for WebSphere MQ section:** Displays MQSeries batch or MQSeries IMS information created by Abend-AID for WebSphere MQ.
- **Language Environment Information section:** Includes information about:
  - *Language Environment run-time options in effect at the time of the error:* Run-time options are obtained from many sources including JCL overrides. These options affect job performance and how a job terminates if an error occurs. Includes how and where the option was set.
  - *User heap storage:* A storage allocation map shows allocated and free storage elements. Overlay errors are identified down to the element in error. Supporting data around the element in error is present to assist in debugging storage overlays.
  - *Significant run-time control blocks (CAA, CIB, EDB, MSIB, PCB, RCB):* In addition to dumping the entire control block, Abend-AID displays important fields in each block, along with the hexadecimal values of those fields.
  - *Run-time library services (RTLS):* Language Environment run-time libraries contain modules that can be accessed via the system LNKLST, STEPLIB, and/or a Language Environment service called RTLS. Sites may choose a single method of access to these Language Environment libraries or allow a combination of these access mechanisms. This varied access can cause many run-time issues which may be the root cause of program failures. Using the information provided in the new Language Environment section of the Abend-AID report, the programmer can now determine at a glance what logical library name and RTLS version were used for the application.
- **Epilog section:** Includes report print and IBM dump information, if applicable.

Extended Language Support

With extended language support you get source code in pertinent sections of the Abend-AID XLS report showing where the fault occurs, listing the conditions under which it occurs, and telling why it occurs. XLS is available for COBOL, PL/I, and Assembler (HLASM) programs.

XLS contributes the following information to the report:

- **Error Analysis section:** Shows the source statement that is in error. For data-related errors, also shows the contents of COBOL fields, PL/I variables, or Assembler variables in the statement at the time of the fault.
- **Error Location section:** Shows the source statement being executed.
- **Program Storage section:** Shows the names and contents of all individual COBOL fields, PL/I variables, or Assembler variables in storage.
- **Program Listing section:** Shows the procedure divisions for COBOL programs on the calling chain or the source listings for PL/I or Assembler programs on the calling chain, with the current statement indicated.

Report Viewing

The Compuware Viewing Facility (Compuware/VF) provides immediate, online access to all report sections. The report can be printed during viewing.

Compuware/VF uses a self-maintaining report dataset that stores Abend-AID XLS output in a compressed format, further reducing DASD space. Reports can be locked to keep them in the report dataset as long as needed.

Compuware/VF keeps track of all Abend-AID XLS activity and provides a comprehensive security system. The report dataset directory summarizes Abend-AID XLS activity in a concise, easy-to-use format. You can sort the directory by job name, report number, error code, date, programmer, or report size.

Source Viewing

Abend-AID XLS's language processors for COBOL, PL/I, and Assembler enable XLS by capturing the pertinent source code from the compiler listing. The language processors store the source in a special, self-maintaining source listing dataset. For XLS, Compuware/VF inserts applicable information from the source listing dataset into the report sections.

The Abend-AID XLS user can set up a language processor to store the source listing in the source listing dataset either before or after an Abend-AID XLS report is written to the report dataset. One or more source listing datasets are prepared by the installer during installation for this purpose. A source listing dataset can be used by other Compuware products such as XPEDITER/TSO.

The COBOL language processor produces an enhanced compiler listing that gives diagnostic information not available in a standard listing. The enhanced listing enables developers to further review program conditions associated with report diagnostics.

Distributed Viewing

Distributed viewing enables application developers to merge Abend-AID XLS report and source listing datasets that are on different MVS images that don't share DASD or to access the datasets individually. Users access the datasets through the Compuware/VF, as they normally would.

Working from individual local systems, any number of users can access report and source listing datasets that both reside on other MVS images. The distance between MVS sites has no affect on the speed or reliability of the extended language report presented to a user.

Database Support

Abend-AID XLS provides comprehensive support for DB2, IMS, and IDMS databases. These database support facilities are available as additional cost options.

Abend-AID for DB2

Abend-AID for DB2 solves difficult S04E and S04F abends, as well as application program problems associated with unexpected negative SQL codes. With Abend-AID for DB2, you no longer need to wade through DB2 dumps and message manuals. Abend-AID XLS pinpoints what happened, as well as when, where, and why a DB2 abend or error occurred. Furthermore, Abend-AID for DB2 usually offers a solution to the problem.

Abend-AID XLS can diagnose DB2 problems regardless of the environment in which the program is executing, and it is applicable to TSO, call, or RRSAF attach mode, IMS, and DL/I attach mode environments. It alleviates the complexity of DB2 as a result of cross-memory services, multiple address spaces, and the different execution modes.

Abend-AID for IMS

Abend-AID for IMS features special diagnostic routines for application failures in IMS DB and TM environments. Abend-AID for IMS reduces the complexity surrounding the IMS environment by giving you comprehensive, easy-to-understand diagnostics on IMS user abends in addition to specific database information. Compuware/VF enables immediate, online access to the Abend-AID XLS report that would not normally be available until the region is brought down.

Abend-AID for IDMS

Abend-AID for IDMS provides diagnostics for failures unique to the CA-IDMS database system. IDMS application failures are thoroughly analyzed by special modules that produce output reports that accurately describe the cause of the error. With Abend-AID for IDMS, you no longer need to rely on manuals or reference cards when debugging IDMS errors.

SNAP-AID

The SNAP-AID facility produces Abend-AID XLS output without terminating the application program. This facility provides a controlled environment for debugging logic or data errors. For example, SNAP-AID can be very beneficial in the DB2 environment when you want Abend-AID XLS diagnostics for a negative SQL code without terminating the program. SNAP-AID is invoked by a call to the SNAPAID module in the application program. The amount of output can be customized by using a parameter list in the SNAP-AID call.

Abend-AID for WebSphere MQ

Abend-AID for WebSphere MQ provides enhanced debugging capabilities for IBM MQSeries batch and IMS applications.

SMF Cost Analysis Tool

This tool analyzes user-provided SMF records spanning any time range to show the volume and type of fault activity on a system. It also estimates the time and system costs saved by using Abend-AID XLS.

Japanese Language Facility

Abend-AID XLS's Japanese language capability uses the IBM double-byte character set (DBCS) to present over 90% of the Abend-AID XLS report in Japanese when viewed online through Compuware/VF. Japanese language is available for DB2, IMS, and COBOL diagnostics. Compuware/VF panels, tutorials, and messages can be displayed in Japanese.

Operation

Abend-AID XLS remains transparent to the user until an application program failure occurs. When a program failure occurs, Abend-AID XLS intercedes and reports the information you need to resolve the problem. Abend-AID XLS operates according to a control structure preset by system defaults. You can tailor Abend-AID XLS's control structure for specialized environments.

Basic Process

When an application program failure occurs in Language Environment, the system calls registered Abnormal Termination Exits (ATE) to process the condition. For abnormal termination processing, Compuware provides the module LEAID as the entry point to Abend-AID XLS.

If the application program failure is an abend, the operating system calls SVC51 to create a dump. In this case, Abend-AID XLS operates by intercepting IBM SVC51 dump processing. If a //SYSUDUMP or //SYSABEND DD statement, or both the //SYSMDUMP and //ABENDAID DD statements are present in the execution JCL, Abend-AID XLS intercepts the call to SVC51 and takes control.

In all Abend-AID XLS processing, Abend-AID XLS modules are dynamically loaded to process the error. These modules must be available either in a link list library or via a JOBLIB or STEPLIB library. Abend-AID XLS tables let you further control Abend-AID XLS execution. Global default options, global routing, and user exits provide a means of customizing Abend-AID XLS processing.

Abend-AID XLS operates in the normal problem-program state rather than the supervisor state. This operation maintains the integrity of the application system.

Abend-AID XLS's decision-making capabilities selectively invoke the necessary diagnostic and reporting modules. This selection is based on variables such as the specific abend code, programming language, and file access method. Abend-AID XLS reports information based on the status of the program and the system at the time of the program failure.

To keep overhead to a minimum, Abend-AID XLS invokes only the required modules for each error. Exact memory requirements depend on the particular error, the options installed, and the complexity of the program being analyzed.

Control Structure

Abend-AID XLS can operate with the default parameters supplied by Compuware or can be customized to your needs. Abend-AID XLS reports can be produced for all application program failures because Abend-AID XLS recognizes which information is most helpful for problem resolution. Abend-AID XLS also provides a sophisticated control structure that can be utilized as needed. Compuware recommends that you employ these controls only after you have identified variations to the product defaults that will aid in resolution of specific types of problems.

Abend-AID XLS offers a seven-level control structure that lets you tailor Abend-AID XLS for specialized environments, such as different levels of support for jobs, programs, or abend codes. The structure is a hierarchical sequence with the highest priority first. Use of a particular control level overrides all of the controls that follow. The seven levels of the control structure are:

1. JCL modification
2. User exits
3. Job and program selection tables
4. Abend code tables
5. Global site options table
6. Global routing criteria and options
7. System-wide control.

Each level is described below.

JCL Modification

This level allows you to easily override the user exits, tables, and default system parameters. Examples of JCL modifications are:

Example 1:

Causes Abend-AID XLS to print the Help page:

```
//ABNLHELP DD DUMMY
```

Example 2:

Causes Abend-AID XLS to format and display all program storage and control blocks, regardless of installation options:

```
//ABNLALL DD DUMMY
```

User Exits

Abend-AID XLS provides optional user exits that you can use to tailor output by particular jobs and/or programs. The exits dynamically perform specialized processing at execution time. They can be particularly useful for program development or for a phased installation of Abend-AID XLS. The exit allows you to:

- Override job and program tables to specify an alternative abend code table for particular errors.
- Route report output to alternate report datasets based on site-specific information, specify that Abend-AID XLS not process for certain jobs or errors, and request the IBM dump either be printed or be suppressed.
- Display selected storage areas, such as GETMAIN storage, or tables and control blocks from user-written applications.

Job and Program Selection Tables

Selection tables let you specify the Abend-AID XLS operation for particular programs or jobs. These operations include *dump* (produces an IBM dump), *nodump* (does not produce an IBM dump), *ignore* (does not produce an Abend-AID XLS report), or *proc* (produces an Abend-AID XLS report).

Abend Code Tables

Tables are supplied with the product that specify — by abend code, PL/I on-code, or LE condition code — the action that Abend-AID XLS will take when a problem is either fully or partially resolved. These actions include options for dump output, control block formatting, and program storage analysis. These tables allow complete flexibility in tailoring the system.

Compuware distributes the abend code tables with the recommended settings. Because these settings are based on Compuware's experience with many installations, modifications to the tables are seldom necessary.

Global Site Options Module — CWGLOBAL

The global site options module (CWGLOBAL) provides various Abend-AID XLS system default options such as amount of information provided in certain sections of the report, language type for report text. This load module is created during installation.

Global Routing Options and Criteria Module — CWROUTE

The global routing options and criteria module (CWROUTE) is created during installation and routes Abend-AID XLS reports to specific report datasets. If specified, criteria can be used to route Abend-AID XLS reports to report databases based on information unique to specific jobs. The names of the default shared directories for reports and source listings are stored here.

System-Wide Control

This level is used to start Abend-AID XLS on your system or to stop Abend-AID XLS from intercepting application program failures.

IBM Dump Provision

In most cases, the Abend-AID XLS report is all that is needed to resolve the problem. In the rare case when it cannot fully resolve the problem, Abend-AID XLS provides all of the information that can be obtained, and then automatically invokes a full IBM dump.

Abend-AID XLS supplies tables that either request or suppress an IBM dump based on abend code, program name, or job name. Abend-AID XLS DD statements can be used to override any option selected with the installation tables.

Utilities

Several Abend-AID XLS utility programs provide users with installation and servicing functions.

- **Installation Verification Report:** Checks the installation of tables and modules.
- **Table Listing Program:** Reviews the contents of the abend code tables, global table with site defaults, job/program selection tables, and the CSECT inclusion table.
- **License Management System:** Installs, checks, or modifies license information.

Other utilities provide users with operational and processing functions.

- **Compuware/VF:** Manages report and source listing datasets online.
- **CSS Utilities:** Sets up a language processor or modifies processing online.
- **CSS Batch File Utilities:** Manages report and source listing datasets in batch mode.

User Manuals

A complete set of user documentation, in several online formats, Adobe Acrobat PDF, HTML, and IBM BookManager is provided on CD-ROM with each release of Abend-AID XLS. Refer to the “Publications” section of the Introduction for complete information.

Chapter 2.
Getting Started

This chapter prepares you to use Abend-AID XLS reports.

Preparing Output

Abend-AID XLS requires a valid //SYSUDUMP DD SYSOUT= statement added to the execution JCL in order to produce output. If you have a //SYSUDUMP DD statement and do not get Abend-AID XLS output, contact an Abend-AID XLS installer.

Note: In Language Environment, a //SYSUDUMP DD statement is required for abends only. It is not necessary for condition code handling.

With Compuware/VF

In order for you to use Compuware/VF (Viewing Facility), Abend-AID XLS output must be routed to an Abend-AID XLS report dataset. This report dataset can be a report shared directory, an individual report database attached to a shared directory, or a "DDIO" report file. Compuware/VF uses the output that is stored in the report dataset.

Normally, your output will be routed to the site-specified default report shared directory. The ABNLTERM DD can be used in a job's JCL to route the Abend-AID XLS report to a specific report dataset.

```
//STEP03 EXEC PGM=PROGNAME
.
.
.
//SYSUDUMP DD SYSOUT=*
//ABNLTERM DD DSN=YOUR.REPORT.FILE,DISP=SHR
```

Before You Begin

This section will help you prepare a sample Abend-AID XLS report to look at in the following section, "Accessing and Browsing a Report" on page 2-5. You will find it of most help in the following situations:

- Your system or site installed Abend-AID XLS for the first time
- Your system or site licensed XLS for the first time
- You are using Abend-AID XLS for the first time.

Extended Language Support

Collect Report Information

The information requested below will be helpful if Abend-AID XLS is new to your site or system or if you are new to XLS. If XLS is new, you may have to refer to an Abend-AID XLS installer.

1. Reports will be automatically routed to a report dataset by the CWROUTE load module. Provide the name of the Abend-AID XLS report dataset that you will use:

2. If report routing is by job name, identify the name required:

3. Identify the Compuware source listing dataset that you will use:

4. Identify how you will access Abend-AID XLS from ISPF, such as with a menu option, CLIST, or other method:

5. Identify the menu, PROC, or JCL member you will use to compile source code with the Compuware language processor:

6. Identify any changes or special parameters required when you use the activating menu, PROC, or JCL member:

Create a Sample Report

Create a sample abend and report for the orientation procedure that follows. You can identify an applicable program containing a sample abend in **Appendix A, "Testing Samples"**. These samples are in the Abend-AID XLS installation library (TPAASAMP). If Abend-AID XLS has been in use on your system or at your site for more than several weeks, you probably have reports available. Or, make a change to a program or alter data to cause the program to abend.

1. Obtain a sample abended program, or create your own, containing a S0C7, on-code 0320, or S0CB.
2. Using the information you recorded in items 5 and 6 above, compile the program incorporating the Compuware language processor.
3. Check the JOBLOG to make sure the Compuware language processor completed with a return code of 0 or 4.
4. Run the program, making sure that it abends.

If routing is by job name, make sure you run the job with an appropriate job name. Refer to item 2 in the preceding section.

Basic Language Support

Collect Report Information

The information requested below will be helpful if Abend-AID XLS is new to your site or system. If Abend-AID XLS is new, you may have to refer to an Abend-AID XLS installer.

1. Verify that Abend-AID XLS reports are being automatically routed to a report dataset by means of the CWROUTE load module created during shared directory installation. Provide the name of the Abend-AID XLS report dataset that you will use:

2. If report routing is by job name, identify the name required:

3. Identify how you will access Abend-AID XLS from ISPF, such as with a menu option, CLIST, or other method:

4. Identify the menu, PROC, or JCL member you will use to compile source code with the Compuware language processor:

5. Identify any changes or special parameters required when you use the activating menu, PROC, or JCL member:

Create a Sample Report

Create a sample abend and report for the orientation procedure that follows. You can identify an applicable program containing a sample abend in **Appendix A**, **"Testing Samples"**. These samples are in the Abend-AID XLS installation library (SORCMAC). If Abend-AID XLS has been in use on your system or at your site for more than several weeks, you have probably have reports available. Or, make a change to a program or alter data to cause the program to abend.

1. Obtain a sample abended program, or create your own, containing a S0C7, on-code 0320, or S0CB.
2. Run the program, making sure that it abends.

If routing is by job name, make sure you run the job with an appropriate job name. Refer to Step 2 in the preceding section.

Accessing Abend-AID XLS

You can view Abend-AID XLS reports using the Compuware/VF (Viewing Facility) either directly under ISPF/PDF or from SDSF in JES2 through a CLIST or REXX EXEC.

Accessing While Viewing JES Output Display

Beginning with Abend-AID XLS Release 9.4 with Compuware Shared Services (CSS) 7.9, a new function allows you to "hot key" from your job log to the report in the Abend-AID XLS report database. You are positioned at the Output Selection Menu (Figure 2-7 on page 2-7). This function uses the same REXX EXEC or CLIST as Compuware/VF (Viewing Facility) that your site's installer set up during CSS installation. Refer to the *Compuware Shared Services User/Reference Guide* for more information. The function is context-independent and works from any ISPF screen display. The function merely depends on a new message, AB400, written at dump capture to the job log.

While viewing the AB400 message, you can do one of the following to access the report:

- Enter the TSO command **TSO EXEC 'AA.SLCCXEXEC(CWVFREXE)' 'HOTKEY'** on the Command Input line (Figure 2-1 on page 2-4). You can define the TSO HOTKEY command as an ISPF system command, invoking Compuware/VF by either CWVFREXE, the REXX EXEC, or CWVFCLSE, the CLIST. The Abend-AID XLS library dataset name (AA.SLCXEXEC) may be different, but the low-level qualifier is typically SLCXEXEC.

- Equate the TSO HOTKEY command to a PF key and thus provide hot-key access to the report (Figure 2-2 on page 2-4). Display the ISPF panel shown in Figure 2-2 on page 2-4 by entering the ISPF **KEYS** command on the command line. This panel enables you to equate a PF key to the command used to link to Compuware/VF. You now have one-key access to the Abend-AID XLS report from SDSF or a SYSOUT archival package.

When you enter the TSO HOTKEY command, the OUTPUT SELECTION MENU (Figure 2-7 on page 2-7) is displayed by Compuware/VF. When you exit the Viewing Facility, the original screen (Figure 2-1) is redisplayed.

Figure 2-1. Executing the TSO HOTKEY Command

```
SDSF OUTPUT DISPLAY AA941C01 J0249094  DSID     2 LINE 10      COLUMNS 02- 81
COMMAND INPUT ===> TSO EXEC 'AA.SLCXEXEC(CWVFREXE)' 'HOTKEY'    SCROLL ===> HALF
11.56.36 J0249094  -AA941C01 RUNTEST  CREATE      00      9      8    .00    .00
11.57.01 J0249094  -AA941C01 RUNTEST  CWAAHOUR    00   2483   3582    .01    .00
11.57.09 J0249094  -AA941C01 RUNTEST  CWAASUBC    00   2646   3829    .01    .00
11.57.10 J0249094  -AA941C01 RUNTEST  LKED        00    468    794    .00    .00
11.57.16 J0249094  +AB400       ===== ABEND-AID =====  013
   013                REPORT WRITTEN TO REPORT NUMBER/DATASET:
   013                000473/EFHDKS0.AA.R94.RPTFILE
   013                JOB-AA941C01 STEP-GO
   013                LE COND CODE=CEE3211S
11.57.17 J0249094  -AA941C01 RUNTEST  GO        3000   1280   4447    .00    .00
11.57.17 J0249094  -AA941C01 RUNTEST  DELETEV     00     49     70    .00    .00
11.57.18 J0249094  -AA941C01 RUNTEST  DELETE      00     39     83    .00    .00
11.57.18 J0249094  IEF404I AA941C01 - ENDED - TIME=11.57.18
11.57.18 J0249094  -AA941C01 ENDED.  NAME-
JCLXLSC3             TOTAL TCB CPU TIME
11.57.18 J0249094  $HASP395 AA941C01 ENDED
```

Figure 2-2. Assign Command to PF Key

```
Command ===>
                                                                More:     +
 Number of PF Keys . . . 24                        Terminal type  . . 3278
 Enter "/" to select . .           (Enable EURO sign)

 PF1 . . . TSO EXEC 'AA.SLCXEXEC(CWVFREXE)' 'HOTKEY'
 PF2 . . . SPLIT
 PF3 . . . END
 PF4 . . . RETURN
 PF5 . . . IFIND
 PF6 . . . RFIND
 PF7 . . . UP
 PF8 . . . DOWN
 PF9 . . . SWAP
 PF10  . . LEFT
 PF11  . . RIGHT
 PF12  . . RETRIEVE
```

Accessing Abend-AID XLS Under an ISPF/PDF Menu

Compuware/VF (Viewing Facility) users can also access Abend-AID XLS through an ISPF/PDF menu, like the one shown below (Figure 2-3 on page 2-5). Your menu selection presents the Compuware/VF Entry Panel. If you cannot access the entry panel from your menu, notify an Abend-AID XLS installer.

Figure 2-3. Sample ISPF/PDF Menu

```
----------------------  ISPF/PDF PRIMARY OPTION MENU  ----------------------

- OPTION  ===> A
                                                              USERID   - USER01
   0 ISPF PARMS   - Specify terminal and user parameters      TIME     - 15:56
   1 BROWSE       - Display source data or output listings    TERMINAL - 3278
   2 EDIT         - Create or change source data              PF KEYS  - 24
   3 UTILITIES    - Perform utility functions
   4 FOREGROUND   - Invoke language processors in foreground
   5 BATCH        - Submit job for language processing
   6 COMMAND      - Enter TSO Command, CLIST, or REXX exec
   7 DIALOG TEST  - Perform dialog testing
   A ABEND-AID    - Abend-AID
   C CSS          - Compuware CSS Utilities
   S SDSF         - System Display and Search Facility
   T TUTORIAL     - Display information about ISPF/PDF
  XT XPEDITER/TSO - Compuware interactive test/debugging product
   X EXIT         - Terminate ISPF using log and list defaults

 Enter END command to terminate ISPF.
```

Accessing and Browsing a Report

The following procedure describes accessing and browsing an Abend-AID XLS report through Compuware/VF.

Note: You can find complete information about the screens referred to in this procedure, including commands and field summaries, in **Chapter 13, "Understanding the Compuware/VF Interface"**.

1. Access the Compuware/VF Entry Panel as described in "Accessing Abend-AID XLS" on page 2-3.
2. At the Compuware/VF Entry Panel (Figure 2-4 on page 2-6), you specify report and source listing datasets. You can also identify the remote MVS images, if applicable, that they're located on. Complete the steps below as applicable, then press **Enter**.

   For extended language support:

   a. Type **0** in the Dataset choice field to select the report dataset. To look only at a compiled source listing instead of a report, specify a source dataset (1 — 8).
   b. If necessary, name the report dataset on the **0** line and one or more source listing datasets on the lines 1 — 8.
   c. If a dataset is located on a different MVS system, in the Remote Server field identify the Abend-AID XLS server on that system. At the Local Server SSID field, identify the Abend-AID XLS server on the local MVS system. Server IDs are assigned when the servers are configured for distributed viewing.

   For base language support:

   a. Type **0** in the Dataset choice field to select the report dataset.
   b. If necessary, name the report dataset on the 0 line.
   c. If a dataset is located on a different MVS system, in the Remote Server field, identify the Abend-AID XLS server on that system. At the Local Server SSID field, identify the Abend-AID XLS server on the local MVS system. Server IDs are assigned when the servers are configured for distributed viewing.

Figure 2-4. Compuware/VF Entry Menu

```
COMPUWARE/VF ENTRY PANEL --------- COPYRIGHT COMPUWARE CORPORATION 1976, 2002.
Command ===>
Dataset choice . 0        (Select 0 - 8, from below, or C for Contact Data)
Member . . . . . ________ (MVS:jobname, SOURCE:program, or member*)
                          (Use the LNAME command to enter a Long Program Name)
                                                                      Remote
       Abend-AID Shared Directory or Report Dataset:         Volume  Server
           0 'AA.REPORTS'
       Source Directory or Listing Datasets:
           1 'AA.LISTING'
           2
           3
           4
           5
           6
           7
           8
Processing options:                                 Source browsing options:
  Confirm delete  . .  YES      (Yes or no)            Unit . . . . . VIO
  Show print setup  .  YES      (Yes or no)            Blocking . . . 10
  Show source warning  YES      (Yes or no, used only for Abend-AID)
  Language  . . . . .  ENGLISH
  Local Server SSID .           (Required for Viewing Remote DDIO files)
```

3. If you're accessing a dataset on a different MVS system, Compuware/VF presents its Alternate Logon screen. Enter your user ID and password for the Abend-AID XLS server on that system in the applicable fields.

Figure 2-5. Compuware/VF Alternate Logon

```
------------------------- COMPUWARE/VF ALTERNATE LOGON -----------------------
 Command ===>




  Logon attempt to AASRV01  requires a USERID/PASSWORD. Please
  supply an alternate userid/password to complete access requirement
  for DDIO file: AA.R900.CXR706.RPTFILE


   Alternate USERID: PFHAJFO     <== USERID on remote system
           PASSWORD:
```

4. At the resulting Abend-AID XLS Dataset Directory (Figure 2-6), select your abended program in the JOBNAME column by entering the line command **S** (Select) to the left of its name. You can apply any of the other commands shown above the directory in the same way.

Figure 2-6. Abend-AID XLS Dataset Directory

```
COMPUWARE/VF ---- 'ABENDAID.REPORT.FILE'-------------------------  ROW 1 OF 52
COMMAND INPUT ===>                                              SCROLL ===> PAGE
                      ABEND-AID DATASET DIRECTORY

      S - View C - Contact Info D - Delete P - Print L - Lock U - Unlock

   JOBNAME   REPT NUMBER JESID   CODE       DATE       TIME    DESC      SIZE(K)
S TSS070CS           71  J08263  SOC7  05 MAR 2000  17.50.22 CLAYTON D   31
  TSS070CS (L)       70  J07553  U1035 05 MAR 2000  16.45.18 DONNA C     15
  TSM050TS           68  J09691  U0240 02 MAR 2000   9.11.49 WOODS       15
  TSM050TS (L)       67  J09465  SOC2  02 MAR 2000   8.51.20 T WOODS     15
  TSM030SN           66  J05446  SNAP  01 MAR 2000  16.33.07 T SMITH     15
  TSM020TX           65  J00942  SOC1  01 MAR 2000   9.31.15 GARY K      15
  TSM050TS           64  J00406  U1020 01 MAR 2000   8.42.12 WOODS TINA  15
  TSM050TS           63  J07030  SOC7  10 FEB 2000  16.54.14 KAYLOR G    31
  TSM050TS (L)       62  J06927  SOCB  10 FEB 2000  16.46.36 WOODS T     31
  TSM030LP           61  J05611  SOC7  10 FEB 2000  15.07.38 SMITH       15
  TSM030PL           59  J09984  S213  10 FEB 2000  14.50.09 T SMITH     15
  TSM050X1 (M)       58  J08064  SD37  08 FEB 2000  11.29.39 TINA WOODS  15
  TSS070CS           57  J08885  SOC7  05 FEB 2000  16.53.20 DONNA C     31
  TSA060RI           55  J00512  U2222 03 FEB 2000  13.54.07 RANDY S     31
  TSA060RO           52  J08065  SOC7  02 FEB 2000  17.36.52 SMITH R     31
  TSM020ST           51  J03476  SOCF  02 FEB 2000  10.19.17 KAYLOR G    15
  TSA060RA           50  J09305  U2222 01 FEB 2000  15.41.46 RANDY S     31
  TSA080PR           46  J05500  SOC7  01 FEB 2000   9.19.48 HEIDI M     15
  TST020LB           45  J06929  SD37  31 JAN 2000  12.52.31 FRANKEN S   15
```

5. At the resulting Output Selection Menu (Figure 2-7), press Enter to go directly to the Error Analysis section. You can directly obtain any report section by entering its identification number or name in the Select Section field.

   If a suitable source listing can't be found *(XLS only),* you will receive a notification screen. The screens are explained in "Source Warnings" on page 2-14.

Figure 2-7. Output Selection Menu

```
COMPUWARE/VF ---------------- OUTPUT SELECTION MENU ----------------------------
SELECT SECTION ===>

'AA.REPORT'        REPT.(L) 66  FLOCON01  J25586

   0 or HEADER   - System, job, and client information
   1 or DIAGS    - Diagnostic information
   2 or NSI      - Next Sequential Instruction (location in program)
   3 or TRACE    - Trace of program flow and program attributes
   5 or REGS     - Register contents and PSWs
   6 or PROG     - Program Storage
   7 or PLIST    - Program Listing (XLS only)
   8 or FILES    - Data management control blocks
   9 or IMS      - IMS diagnostic information and PCBs
  10 or IDMS     - CA-IDMS diagnostic and status information
  11 or DB2      - DB2 diagnostic and status information
  12 or SORT     - Current COBOL sort information
  13 or DATALOC  - COBOL Data Locator
  14 or MQ       - WebSphere MQ diagnostic information
  15 or LE       - LE information
  16 or EPILOG   - Abend-AID termination information

               Key section number or name and press ENTER.
    Or press ENTER to start at top of output and view sections 1 through 5.
```

6. The Error Analysis section shows the text of the abended statement (*XLS only*) and information about all fields in the statement. Scroll up to see the header section.

Figure 2-8. Error Analysis Section

```
COMPUWARE/VF ------------------------------------------------------------------
COMMAND INPUT ===>                                          SCROLL ===> CSR

'ABENDAID.REPORT.FILE'  REPT.71  AA940XLS  J08263

   A data exception, S0C7, occurred during execution of program PAYROLLX.

   ******************************************
   *          Analysis of Error             *
   ******************************************

   A Data Exception occurred in the following statement:

 000115              SUBRACT TCR-TOTAL-TAXES FROM TCR-GROSS-PAY
 000116                 GIVING WS-NET-PAY


                   Current values of fields on abending statement
          Level/Field Name                    Picture/Type Offs      Value
          ----------------------------------  ----------------- ----+----1----
 000009   02 TCR-TOTAL-TAXES                    9(4)V99           013222
 000010   02 TCR-GROSS-PAY                   # 9(2)V99         X'7C7CF3F3F1F1F0'
 000012   77 WS-NET-PAY                      # 9(2)V99 COMP-3  X'CC33110F'
 000015   02 RPT-GROSS-PAY                     $ZZ,ZZ.99         $ 1,203.33
          ------------------------------------------------------------------------
```

Note: You can activate these functions at the COMMAND INPUT field within the report:

- Obtain any report section by entering its menu number or name.
- Search for specific data with the F (Find) command.
- Suppress source information by entering Source On and Source Off (*XLS only*).

For complete information about functions available within the report, refer to "Output Selection Menu" on page 13-9 or "Report Commands" on page 2-13.

7. Sections after the Registers section can be selected individually at the COMMAND INPUT field. Enter **6** on the command input line to see the Program Storage (PROG) section.

Figure 2-9. Trace Section

```
COMPUWARE/VF ------------------------------------------------------------------
COMMAND INPUT ===>                                          SCROLL ===> CSR

'ABENDAID.REPORT.FILE'  REPT.71  AA940XLS  J08263

   ****************************************
   *          Call Trace Summary          *
   ****************************************

    ******Calling*****     ****Return****          ******Called******
    Load-Mod  Program      Type     Value          Program   Load-Mod

    *SYSTEM                                 Links to PAYROLLX  PAYLOAD *

    PAYLOAD   PAYROLLX     DISP  000007A2  Calls    IGZCSPA   IGZCPAC

                                          Program Causing Error  *

   ------------------------------------------------------------------------
   Load-Mod    Amode  Rmode  Resides in

   PAYLOAD     ANY      24    CW.AA.TEMPLOA7
   IGZCPAC     24       24    SYS2.COBRUN
   ------------------------------------------------------------------------

   ****************************************
   *     Application Program Attributes     *
   ****************************************

   Load-Mod   Program     Compile Date  Length     Language

   PAYLOAD    PAYROLLX    08 FEB 2002   00000FB8   VS COBOL II V1R3M2
```

8. If more than one program was on the calling chain at the time of the abend, you will see a Program Storage Selection List (Figure 2-10). Select a program by entering **S** to the left of the program name.

Figure 2-10. Program Storage Selection List

```
COMPUWARE/VF PROGRAM STORAGE SELECTION LIST  ----------------  ROW 1 TO 2 OF 2
COMMAND INPUT ===>                                           SCROLL ===> PAGE

PROGRAM / PROCEDURE NAME           LOAD   LANGUAGE      COMPILE DATE  -TIME

S PAYROLLX                      PAYLOAD  VS COBOL II   15 FEB 2000  11:23:36
  UPDATE                        PAYLOAD  VS COBOL II   15 FEB 2000  11:24:10
****************************** BOTTOM OF DATA ******************************
```

9. You will see the Working-Storage and Linkage sections of the program (Figure 2-11 on page 2-10). You may have to press the Down key once or twice to bypass the compiler options information. Enter **7** on the command line to see the Program Listing (PLIST) section.

Figure 2-11. Program Storage Section (Working Storage)

```
COMPUWARE/VF-----------------------------------------ROW 39 TO 57 OF 163
COMMAND INPUT ===> 7                                   SCROLL ===> PAGE

'ABENDAID.REPORT.FILE'  REPT.71  AA903XLS  J08263

               Working-Storage Section

 Working storage referenced by BLW cell      0 (X'0')

 Level/Field Name                  Picture/Type Offs      Value
 --------------------------------  ---------------- ----+----1----+----2
 77 WS-RECORD-CTR                  S999     COMP-3   +4
 77 WS-TOTAL-NET-PAY               S9(7)V99 COMP-3   +339117
 77 WS-NET-PAY                     S9(5)V99 COMP-3 X'CC33110F'
 77 WS-EMP-NO                      9(8)              04060293
 01 WS-TIMECARD-REC
  02 TCR-EMP-NO                    9(8)              04060293
  02 TCR-MONTH-CODE                99                02
  02 TCR-EMP-NAME                  X(32)             EDWARDS, ALEXANDER
  02 TCR-GROSS-PAY                 9(5)V99         X'7C7CF3F3F1F1F0'
  02 TCR-TOTAL-TAXES               9(4)V99           013222
```

10. If more than one program is available, you will see a Program Listing Selection List similar to the Program Storage Selection List in Figure 2-10 on page 2-9. Select a program by entering **S** to the left of the program name.
11. You will see the paragraph containing the current statement (Figure 2-12). The command PLIST ALL shows the complete division or listing. To return to the Output Selection Menu, press the End key. You may have to press it more than once.

Figure 2-12. Program Listing Section

```
COMPUWARE/VF-----------------------------------------ROW 1 TO 24 OF  24
COMMAND INPUT ===>                                     SCROLL ===> PAGE

'ABENDAID.REPORT.FILE'  REPT.71  AA903XLS  J08263

    ************************************************************************
    *              Program Listing Section - PAYROLLX                      *
    ************************************************************************

 000100        READ-A-RECORD.
 000101            READ TIMECARD INTO WS-TIMECARD-REC
 000102                AT END GO TO END-OF-JOB.
 000103            MOVE WS-TIMECARD-REC TO MASTER-RECORD.
 000104            ADD 1 TO WS-RECORD-CTR.
 000105            IF TCR-EMP-NO > WS-EMP-NO
 000106                MOVE TCR-EMP-NO TO WS-EMP-NO
 000107            ELSE
 000108                DISPLAY 'EMPLOYEE# SEQ ERROR  ',
 000109                        TCR-EMP-NO,
 000110                        SPACE,
 000111                WS-EMP-NO UPON SYSOUT,
 000112                GO TO READ-A-RECORD.
 000113            MOVE TCR-EMP-NO TO RPT-EMP-NO.
 000114            MOVE TCR-EMP-NAME TO RPT-EMP-NAME.
```

12. If your program has open files, you may want to view the File section by entering **8**.

Figure 2-13. Output Selection Menu

```
COMPUWARE/VF ---------------- OUTPUT SELECTION MENU ---------------------------
SELECT SECTION ===>

'AA.REPORT'        REPT.(L) 66  FLOCON01  J25586

   0 or HEADER   - System, job, and client information
   1 or DIAGS    - Diagnostic information
   2 or NSI      - Next Sequential Instruction (location in program)
   3 or TRACE    - Trace of program flow and program attributes
   5 or REGS     - Register contents and PSWs
   6 or PROG     - Program Storage
   7 or PLIST    - Program Listing (XLS only)
   8 or FILES    - Data management control blocks
   9 or IMS      - IMS diagnostic information and PCBs
  10 or IDMS     - CA-IDMS diagnostic and status information
  11 or DB2      - DB2 diagnostic and status information
  12 or SORT     - Current COBOL sort information
  13 or DATALOC  - COBOL Data Locator
  14 or MQ       - WebSphere MQ diagnostic information
  15 or LE       - LE information
  16 or EPILOG   - Abend-AID termination information

               Key section number or name and press ENTER.
    Or press ENTER to start at top of output and view sections 1 through 5.
```

13. If more than one file is open at the time of abend, you will see a list of all open files on the File Summary Selection List (Figure 2-14). Select the file you want to view by entering **S** to the left of the file name. To open a file through File-AID's edit or browse modes, enter **E** or **B**.

Figure 2-14. File Summary Selection List

```
COMPUWARE/VF  FILE SUMMARY SELECTION LIST  ------------------  ROW 1 TO 8 OF 8
COMMAND INPUT ===>                                          SCROLL ===> PAGE

  S - view file control blocks  E - File-AID Edit  B - File-AID Browse
                                (except JES spooled datasets)

  DDNAME  DEVICE UNIT VOLSER DISP AMTYPE  EXCPS   ------ DCB INFORMATION ---

  OUTLIST  JES2 spooled            QSAM   81       (DSORG=PS,RECFM=FBA,
                                                   LRECL=133,BLKSIZE=13300)
  MASTER   3380  3AB  CWX018 NEW   QSAM    0       (DSORG=PS,RECFM=FBA,
                                                   LRECL=80,BLKSIZE=3200)
S TIMECARD                         VSAM    1
  SYSUT1   3380  2C9  SMS801 SHR   VSAM    4
  SYSUT1   3380  2C9  SMS801 SHR   BDAM    4       (DOSORG=DA,RECFM=F,
                                                   LRECL=121,BLKSIZE=12100)
****************************** BOTTOM OF DATA ************************
```

14. Press the Down key to view the complete file information (Figure 2-15 on page 2-12). Press the End key to return to the File Selection Menu. Press End again to return to the Output Selection Menu.

Figure 2-15. File Section

```
COMPUWARE/VF ------------------------------------------- ROW 1 TO 24 OF 36
COMMAND INPUT ===>                                        SCROLL ===> PAGE

                              Use the EDIT or BROWSE command for File-AID

'ABENDAID.REPORT.FILE'  REPT.71  AA903XLS  J08263

       Data Management Control Blocks for DDNAME - TIMECARD

       DSNAME=CWV.TSO000.AA.TIMECARD
                                                              ACC METH=VSAM

  File EXCP Count=1

  File Summary:

  Access type........BASE CLUSTER
  Dataset type.......ESDS
  Processing type....ADR,AMODE31,NFX,DDN,NDF,SEQ,NCI,IN,NIS,NRM,NRS,NSR,
                     NUB

  File errors........None

  Current record:

  00044F70   CHAR  4058829912DOE,  WILLIAM  T.               00|@#200
```

15. Exit Compuware/VF by entering **End** at Select Section.

Figure 2-16. Output Selection Menu

```
COMPUWARE/VF ---------------- OUTPUT SELECTION MENU ---------------------------
SELECT SECTION ===>

'AA.REPORT'        REPT.(L) 66  FLOCON01  J25586

   0 or HEADER   - System, job, and client information
   1 or DIAGS    - Diagnostic information
   2 or NSI      - Next Sequential Instruction (location in program)
   3 or TRACE    - Trace of program flow and program attributes
   5 or REGS     - Register contents and PSWs
   6 or PROG     - Program Storage
   7 or PLIST    - Program Listing (XLS only)
   8 or FILES    - Data management control blocks
   9 or IMS      - IMS diagnostic information and PCBs
  10 or IDMS     - CA-IDMS diagnostic and status information
  11 or DB2      - DB2 diagnostic and status information
  12 or SORT     - Current COBOL sort information
  13 or DATALOC  - COBOL Data Locator
  14 or MQ       - WebSphere MQ diagnostic information
  15 or LE       - LE information
  16 or EPILOG   - Abend-AID termination information

               Key section number or name and press ENTER.
    Or press ENTER to start at top of output and view sections 1 through 5.
```

Report Commands

You can scroll through the first five report sections. Other sections must be accessed individually. Enter the section number or name in the COMMAND INPUT field. Inside the report, use standard scrolling commands.

The commands summarized below are also available.

Source Command

You can suppress the presence of source information in the report or look at a directory listing of the members in your source datasets.

Enter SOURCE in the COMMAND INPUT field of any report section containing source information or in the SELECT SECTION field of the Output Selection menu.

Source On

Default. Display source. Applicable only to the open report.

Source Off

Suppress source.

Source Mismatch

When you're working with COBOL or PL/I XLS, this command presents a source warning screen at which you can access a directory listing of the source datasets specified on the Entry Panel. For Assembler XLS, it directly presents the directory listing.

F (Find) Command

When viewing a report section, use the F (Find) command to search for specific characters or character strings within the section:

Note: The word *Find* cannot substitute for any of the following syntax.

F xxx

Finds the characters specified by **xxx** in the current report section. Ignores letter case.

F 'xxx yyy'

Finds the character string specified by '**xxx yyy**' with embedded blanks. Ignores letter case.

F

Finds the character string specified in the previous FIND command. Ignores letter case. Use this command instead of the <Repeat Find> key.

FS xxx

Finds the characters specified by **xxx** in the current report section. Case-sensitive.

FS 'xxx yyy'

Finds the character string specified by **'xxx yyy'** with embedded blanks. Case-sensitive.

FS

Finds the character string specified in the previous FIND command. Case-sensitive. Use this command instead of the <Repeat Find> key.

Source Warnings

You receive a source warning if the following condition is in effect:

- The source listing datasets specified on the Entry Panel do not contain a source listing that matches the program identified in the selected report.

When you recompile your program to create a listing to be placed in the source listing dataset, several factors can affect the matching of the re-created listing with the original program object module. These factors include the following:

- The source program and any included source code must be the same.
- Compiler levels must be the same.
- Compiler options affecting object code generation, such as OPTIMIZE, must be the same.

As a result, you may encounter the following warning screens.

COBOL Mismatched Date and Time

When the compile date and time of a COBOL program being diagnosed does not match a program in the source listing dataset, you receive the Mismatched Date and Time Warning, shown in Figure 2-17. You also obtain this screen with the SOURCE MISMATCH command.

Figure 2-17. COBOL Mismatched Date and Time

```
COMPUWARE/VF ----------------------------------------------------------------
COMMAND INPUT ===>

'PFHKXLO.CSS79.QA.REPTFILE'  REPT.4  EFHRIPOC  J31098

     * * * * * * * * * * * *    W A R N I N G    * * * * * * * * * * * *
     *       Program                            Most recent listing      *
     *       ------------------------------------------------------------ *
     * Name: COBSAMP                            COBSAMP                  *
     * Load: COBSAMP                                                     *
     * Date: 07 DEC 2000                        27 DEC 2001              *
     * Time: 11:22:20                           08:07:38                 *
     *                                                                   *
     * The program date and time stamp does not match a source listing.*
     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 Type an S next to the desired option and press ENTER.

 _ Use the MOST RECENT source listing shown above.
 _ Display available source listings.
 _ Retrieve a source listing by submitting background JCL.
 _ Provide BASIC support without a source listing.

 Enter END command to exit.
```

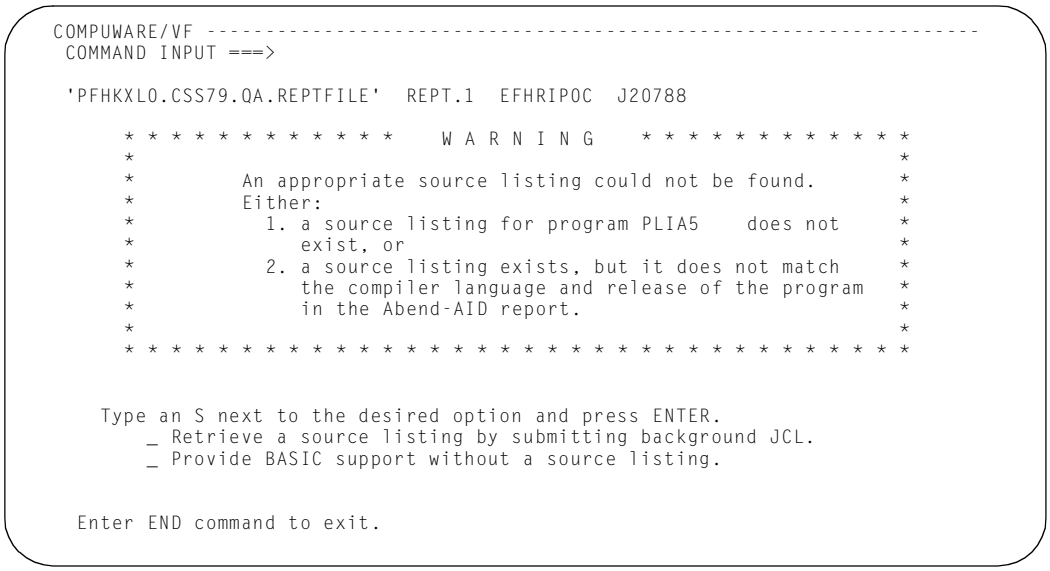
From this screen, you can choose one of the following:

- Continue with the most recent available source listing.
- Display available source listings.
- Submit a background job to retrieve a source listing.
- Continue without source support.
- Exit the selected section.

PL/I Mismatched Structure

When the compile time of a PL/I program is not in the load module, Abend-AID XLS can't report it. A mismatch results. When a PL/I date or time mismatch occurs, Abend-AID XLS automatically checks the most recent source listing for matching program structure. If the structures do not match, you receive the Mismatched Structure Warning. You also obtain this screen with the SOURCE MISMATCH command.

Figure 2-18. PL/I Mismatched Source Listing Warning



From this screen, you can choose one of the following:

- Submit a background job to retrieve a source listing.
- Continue without source support.
- Exit the selected section.

Assembler Unmatched Source Listing

Due to the characteristics of Assembler, Abend-AID XLS may not be able to reliably identify a matching source listing dataset for Assembler diagnostics. In this case you will receive a directory similar to the one shown in Figure 2-19 on page 2-16. Select the member you need.

The directory is immediately presented when Abend-AID XLS can't identify a matching source listing for Assembler diagnostics. In a diagnostic report for Assembler, you can also obtain this directory with the SOURCE MISMATCH command.

Figure 2-19. Assembler Mismatched Warning

```
COMPUWARE/VF ------------------------------------------  ROW 1 TO 28 OF 31
 COMMAND INPUT ===>                                       SCROLL ===> CSR

     CSECT Name: EUR001
  Wildcard Mask: ********
      Assembled: 16 AUG 2000            Load module: EUR001

    The CSECT name and date do not uniquely match a source listing.

  _ Retrieve a source listing by submitting background JCL.
  _ Provide BASIC support without a source listing.

    Type an S next to the correct source listing below and press the ENTER key.
                ASSEMBLER SOURCE LISTING DATASET DIRECTORY
    PROGRAM    LIST NUMBER     ASMB DATE     TIME     RC   LANGUAGE    SIZE(K)
   ---------- 'EFHAXS0.TEST.DDIO' ----------------------------------------
   X80102B   (L)       3537    09 NOV 2001  12.20     00   HLASM R4      399
   TEMPNAME  (L)       3344    10 SEP 2001  13.33     00   HLASM R4      160
   TEST1     (L)       3179    22 JUN 2001  11.35     00   HLASM R4      160
   VIBAIITF  (L)       3046    28 MAR 2001  13.07     00   HLASM R4      559
   CWDEMASM  (L)       2612    14 JUN 2000  20.08     04   HLASM R3      240
   ASM03T    (L)       2651    08 JUN 2000  15.47     00   HLASM R3       80
   ASMPRG1   (L)       2330    13 MAR 2000  12.14     00   HLASM R3       80
   AIBHLP00  (L)       1023    T PAGO N     10.51     00   HLASM R2      719
```

Available Source Listings

If you choose Display Available Source Listings in response to mismatch warnings for COBOL and PL/I diagnostics, you receive a Dataset Directory similar to the one shown in Figure 2-20.

You can use these commands in the directory:

B: Browse.

D: Delete.

L: Lock.

P: Print.

S: Select.

U: Unlock.

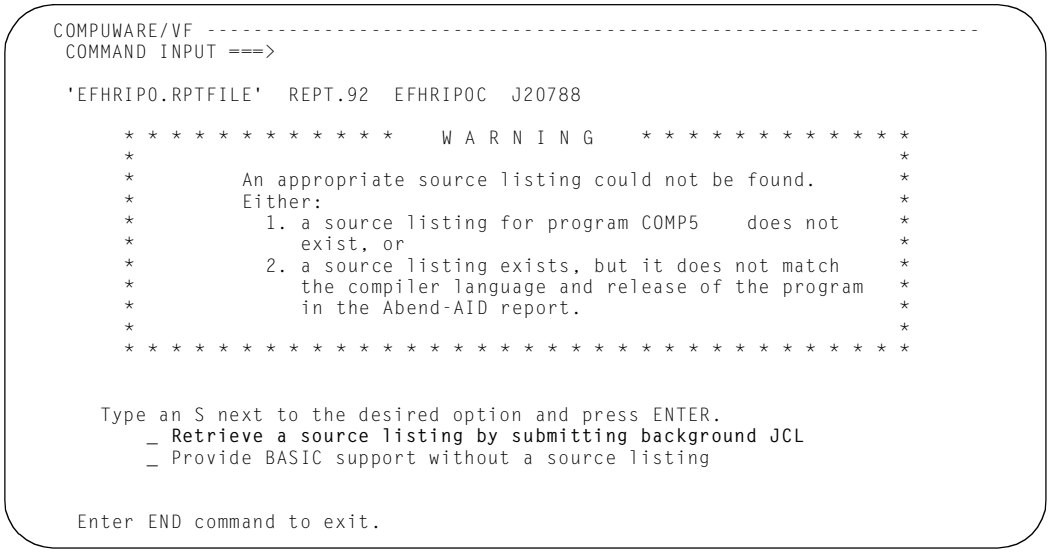
Figure 2-20. Source Listing Dataset Directory

```
COMPUWARE/VF ------------------------------------------  ROW  1 TO 13 OF 13
COMMAND INPUT ===>                                       SCROLL ===>  PAGE
                  SOURCE LISTING DATASET DIRECTORY
   PROGRAM    LIST NUMBER   COMP DATE    TIME     RC   LANGUAGE    SIZE
  ---------- 'CWX0236.TECH.FIL1169.SLS2' --------------------------------
  WTP#1074  (L)        16   05 MAY 2000  16:16:18 00   PL/I V2    48 KB
  WTP#1074              8   29 MAR 2000  09:16:26 00   PL/I V2    32 KB
  ---------- 'CWX0236.TECH.FIL1169.SLS1' --------------------------------
  WTP#1074  (L)        17   04 MAY 2000  10:49:45 00   PL/I V2    82 KB
  WTP#1074             16   03 MAR 2000  09:16:26 00   PL/I V2    82 KB
  ---------- 'CWX0236.BD60.SLSP' ----------------------------------------
  WTP#1074  (L)        19   05 MAY 2000  16:16:18 00   PL/I V2    44 KB
  WTP#1074  (L)        12   04 FEB 2000  10:49:45 00   PL/I V2    40 KB
  WTP#1074  (L)        10   02 FEB 2000  09:16:26 00   PL/I V2    32 KB

****************************** BOTTOM OF DATA ******************************
```

Source-Not-Found

You receive the warning shown in Figure 2-21 when the program name and language identified in the report can't be found in a specified source listing dataset.

Figure 2-21. Source Not Found Warning



From this screen, you can choose one of the following:

- Continue without source support.
- Exit the selected section.

Rapid Response

Abend-AID XLS is one of the *find-and-fix* tools in Compuware's Rapid Response survival kit. In addition to diagnostic reporting at the time of abend, several Abend-AID XLS facilities enable applications programmers to respond as quickly as possible to program faults that may be associated with Euro-currency program changes. These facilities provide advantages in development and testing stages, as well as during production.

Abend-AID XLS's Rapid Response advantages, summarized below, require Compuware Shared Services Release 7.8 or more current. For complete information, refer to the *Compuware Shared Services User/Reference Guide* or the Compuware/VF online help. You can also find additional information about the report Data Locator section in "COBOL Data Locator" on page 3-5.

File-AID Access

Abend-AID XLS users can keystroke directly into a Compuware File-AID product from an Abend-AID XLS report:

- From the File section you can access File-AID/MVS.
- From the DB2 section you can access File-AID/DB2.
- From the IMS section you can access File-AID/IMS.

Edit and Browse commands in the report take you into File-AID edit and browse modes within a dataset or database. A current version of the applicable File-AID product must be available on the operating system.

Data Locator

An online report section for COBOL diagnostics presents all data-related source and fields, including Millennium Language Extensions modifications if MLE is active. Selecting COBOL Data Locator on the output menu shows you all formats used and flags data or formats that may be invalid.

Access the Data Locator section as you would any report section. Data locator specifications may be customized for any data strings or fields.

Source Comparison

A Compuware Shared Services (CSS) batch utility command enables you to compare any two source listings in a source listing dataset. The Flag command identifies any lines in a more recent listing that differ from a previous listing. You can also compare listings in different source listing datasets.

Contact Information

You can attach a contact name and additional information to an abended job for future reference. Access the Contact Information summary from the Abend-AID XLS Primary Menu, the Entry Panel, or the Abend-AID XLS Dataset Directory.

CSS Utilities

With the CSS Utilities you can modify compile JCL to preprocess or postprocess source code for XLS, or create and format report and source listing shared directories and associated files, all by entering specifications on ISPF panels. These operations previously involved JCL preparation and were typically done by a systems programmer. Access the utilities through the Abend-AID XLS Primary Menu.

Chapter 3.
Diagnosing a COBOL Data-Related Error

This chapter describes how to use Abend-AID XLS to diagnose a common COBOL error. Abend-AID XLS provides two levels of COBOL support: extended language support and basic language support.

Resolving an S0C7 with Extended Language Support

When you select a report section online through the Compuware/VF Viewing Facility (Compuware/VF), with the applicable Compuware language processor in use, program source code can be automatically merged into the report. This comprises Abend-AID XLS (Extended Language Support).

COBOL XLS provides:

- The source code of the statement in error, shown in the Error Analysis section. For data-related errors, all COBOL fields in the statement in error are shown with their attributes and current values.
- The source code of the statement being executed, shown in the Error Location section.
- The names and contents of all individual COBOL fields in working storage, shown in the Program Storage section.
- The procedure divisions for COBOL programs on the calling chain, shown in the Program Listing section with the current statement indicated.

The following procedure shows how to read an extended language report on a common error, an S0C7 abend.

1. Review the Error Analysis section to determine the cause of the error.

   First review the Error Analysis section, shown in Figure 3-1 on page 3-2, to find the cause of the error. Usually, all the information you need is in this section. The source code of the statement in error is shown, and all fields are shown with their current values. The Error Analysis section also gives a definition of a data exception. In this example, fields WS-NET-PAY and TCR-GROSS-PAY both contain invalid data, indicated with the # (pound sign) character.

Figure 3-1. Error Analysis Section

```
   A data exception, S0C7, occurred during execution of program PAYROLLX.

   ****************************************
   *          Analysis of Error           *
   ****************************************

   A Data Exception occurred in the following statement:

000115              SUBTRACT TCR-TOTAL-TAXES FROM TCR-GROSS-PAY
000116                  GIVING WS-NET-PAY

                  Current values of fields on abending statement
          Level/Field Name                    Picture/Type Offs      Value
          ----------------------------------  ----------------- ----+----1----
000012    77 WS-NET-PAY                      # 9(2)V99 COMP-3  X'CC3311OF'
000010    02 TCR-GROSS-PAY                   # 9(5)V99           @@3311O
000009    02 TCR-TOTAL-TAXES                   9(4)V99           013222
          ------------------------------------------------------------------------
                  "#" - Indicates field contains invalid data
   The field causing the exception is located in Working-Storage of program
   PAYROLLX.

   Because the program is re-entrant, Working-Storage is located outside of
   the program.

   ------------------------------------------------------------------------
   A Data Exception is caused when a computational-3 field has
   an invalid digit (not 0-9), or its last byte contains an invalid
   sign (not A, B, C, D, E, or F).
```

2. Review the Error Location section to find the location of the error.

   The Error Location section, shown in Figure 3-2, indicates that the statement being executed is contained in paragraph READ-A-RECORD of program PAYROLLX.

   The Error Location section also provides the following information:

   – Program's compile date and length
   – Program's link date and load module length
   – Load module name and the load library name
   – Location of the last I/O operation or subroutine call, if applicable.

Figure 3-2. Error Location Section

```
  ****************************************
  *             Error Location           *
  ****************************************

  The statement being executed was:

000115              SUBTRACT TCR-TOTAL-TAXES FROM TCR-GROSS-PAY
000116                  GIVING WS-NET-PAY

  This statement is contained in
  paragraph "READ-A-RECORD" in program "PAYROLLX".

  The program was compiled on 08 FEB 2002 and is 00000FB8 bytes long.

  It is part of load module PAYLOAD.

  The module was loaded from STEPLIB library CW.AA.TEMPLOA7.

  The module was link edited on 08 FEB 2002 and is 00001400 bytes long.

  The last known I/O request or call in the program above was:

000101            READ TIMECARD INTO WS-TIMECARD-REC.

  This statement is contained in
  paragraph "READ-A-RECORD" in program "PAYROLLX".
```

The Error Analysis and Error Location sections usually provide enough information to resolve the problem. The other sections of the Abend-AID XLS report provide supporting information that can be used to trace the cause of the error.

3. Find the contents of other fields in your program.

   To find the contents of other data fields, review the Program Storage section in the Abend-AID XLS report. Program storage is formatted by individual field name within working storage and linkage storage. Level name, field name, picture definition, and contents at time of error are given, as shown in Figure 3-3.

   To find the contents of an indexed field, refer to "Determining Index and Indexed Field Values" on page 3-4.

Figure 3-3. Program Storage Section

```
             Working-Storage Section
Level/Field Name                    Picture/Type Offs      Value
----------------------------------  ----------------- ----+----1----+----2
77 WS-LINE-CTR                      S999     COMP-3   +4
77 WS-PAGE-CTR                      S999     COMP-3   +1
77 WS-RECORD-CTR                    S999     COMP-3   +4
77 WS-TOTAL-NET-PAY                 S9(7)V99 COMP-3   +339117
77 WS-NET-PAY                     # S9(5)V99 COMP-3 X'CC3311OF'
77 WS-EMP-NO                        9(8)              04060293
01 WS-TIMECARD-REC
 02 TCR-EMP-NO                      9(8)              04060293
 02 TCR-MONTH-CODE                  99                02
 02 TCR-EMP-NAME                    X(32)             EDWARDS, ALEXANDER
 02 TCR-GROSS-PAY                 # 9(5)V99         X'7C7CF3F3F1F1F0'
 02 TCR-TOTAL-TAXES                 9(4)V99           013222

             Linkage Section
Level/Field Name                    Picture/Type Offs      Value
----------------------------------  ----------------- ----+----1----+----2
01 PARMDATA
 02 PARM-LEN                        9(3)     COMP     7
 02 PARM-STRING                     X(7)              SPACES
-------------------------------------------------------------------------------

                              Supporting data
                           Contents of BLW cells
    BLW      0 (X'0')    - 00007220

                           Contents of BLL cells
    BLL      0 (X'0')    - 00000000   BLL      1 (X'1')    - 00013808

                           Contents of BLF cells
    BLF      0 (X'0')    - 09BF8270   BLF      1 (X'1')    - 00007161
    BLF      2 (X'2')    - 09BF5030

                           Task Global Table (TGT)
 DSPL Address   --------------- Data ---------------
00000 00006B28  00108001 00015E98 00000000 80006A52  ......;q......].
00010 00006B38  89BD52D8 8004C3DC 00DD01E0 00000001  i..Q..C....\....
00020 00006B48  00006F60 000177EC 00006D00 00006FC0  ..?-......_...?{
00030 00006B58  80006860 09BF8270 09BF4FE0 00007161  ...-..b...|\.../
00040 00006B68  0000680C 000064BC F3E3C7E3 00026018  ........3TGT..-.
```

4. Review other supporting information.

   Whenever COBOL source listings are available in the source listing dataset, XLS provides the procedure divisions for all active COBOL programs with the current statement indicated. This information is shown in the Program Listing section. Figure 3-4 on page 3-4 shows the program listing for PAYROLLX.

Figure 3-4. Program Listing Section

```
    ****************************************************************
    *              Program Listing Section - PAYROLLX              *
    ****************************************************************


000100        READ-A-RECORD.
000101            READ TIMECARD INTO WS-TIMECARD-REC
000102                AT END GO TO END-OF-JOB.
000103            MOVE WS-TIMECARD-REC TO MASTER-RECORD.
000104            ADD 1 TO WS-RECORD-CTR.
000105            IF TCR-EMP-NO > WS-EMP-NO
000106                MOVE TCR-EMP-NO TO WS-EMP-NO
000107            ELSE
000108                DISPLAY 'EMPLOYEE# SEQ ERROR  ',
000109                        TCR-EMP-NO,
000110                        SPACE,
000111                WS-EMP-NO UPON SYSOUT,
000112                GO TO READ-A-RECORD.
000113            MOVE TCR-EMP-NO TO RPT-EMP-NO.
000114            MOVE TCR-EMP-NAME TO RPT-EMP-NAME.
 CURR STMT ==>    SUBTRACT TCR-TOTAL-TAXES FROM TCR-GROSS-PAY
000116                GIVING WS-NET-PAY
000117            MOVE TCR-GROSS-PAY TO RPT-GROSS-PAY.
000118            MOVE TCR-TOTAL-TAXES TO RPT-TOT-TAXES.
000119            MOVE WS-NET-PAY TO RPT-NET-PAY.
000120            ADD WS-NET-PAY TO WS-TOTAL-NET-PAY.
```

5. Review record information.

   The File section shows information about all files that are open at the time of the error. This information includes identification of the current record and previous records, when available. DCB and other control block information is also given. Abend-AID XLS displays records using decimal locations, making fields easier to find when using record layouts. File information is provided for all file types, including VSAM and IAM.

   You can directly access these files through File-AID from the online File Summary Selection List, which lists multiple open files. Inside a file you have full File-AID edit and browse capabilities.

Searching Program Listing

The commands shown below enable you to immediately access the source statements that you need, if you're working with XLS. Enter them as shown in an open Program Listing section with the PLIST command.

blank

Displays source statements as determined by the SLSLIST or SLSMLIST option in the CWGLOBAL table.

nn

Displays *nn* lines before and after the current statement.

ALL

Displays the complete COBOL procedure division.

PARA

Displays the COBOL paragraph containing the current statement.

Determining Index and Indexed Field Values

The following procedure tells how to determine an indexed occurrence within a table using an extended language report.

1. If the index is named in the statement in error, review the Error Analysis section. Otherwise, review the Program Storage section for your program.
2. Find the index name below the corresponding table name. Make note of the index value.
3. To convert the index value to the actual occurrence value, do the following:
   a. Divide the index value by the length of one occurrence within the table. For example, if the index value is 27 and the length of one occurrence is 9: 27 / 9 = 3.
   b. Add one to this value: 3 + 1 = 4.

   The result is the occurrence within the table that the index was pointing to when the Abend-AID XLS report was created.

COBOL Data Locator

With default settings, the selection COBOL Data Locator on the output menu provides a summary of all data-related source and data formats in use, identifying data or formats that may be invalid. The summary includes Millennium Language Extensions modifications if MLE is active at time of abend. The locator's search and display specifications may be customized for any types of data, including dates and currency.

The summary is available only online, for COBOL programs. It may be printed during viewing with the LPRINT command, but does not print out with the complete Abend-AID XLS report.

The data locator summary is shown as a Working Storage report section. When it's open, you can access locator search specifications by entering FINDDATA on the command line. When you exit after changing specifications, the summary will be immediately refreshed with the specified data.

For detailed information, refer to the Compuware/VF online help or the Compuware/VF chapter of the *Compuware Shared Services User/Reference Guide*.

Figure 3-5. COBOL Data Locator Summary

```
COMPUWARE/VF ----------------------------------------------------------------
COMMAND INPUT ===> _                                          SCROLL ===> CSR
                                                  Formatting has completed
'PFHWFBO.DUMP.DDIO'      REPT.71   PFHWFBV2  JO6619
 Program: MLECOB          Compile Date: 07 JAN 2000    YEARWINDOW(1971)

 '@' in column    36 denotes field with Data Locator PICTURE match.
 '!' in column    36 denotes field with Data Locator DATE    match.
 '#' in column    36 denotes field contains data not matching PICTURE clause.

   Language Processor Interface Release      = 07.07.00
   Language Processor Post Processor Release = 07.07.00


            Working-Storage Section

Level/Field Name               Picture/Type Offs        Value
-----------------------------  -----------------   ----+----1----+----2
77 CURRENT-YEAR              #  99                   SPACES
77 DAT-DATA-MATCH            !  9(5)                 99366
77 DUE-YEAR                     99                   05
              DATE FORMAT    YY       YEAR(2005)
77 END-YEAR                     9999                 2005
              DATE FORMAT  YYYY       YEAR(2005)
77 EXP-CUST-DATE             @  9(8)                 19580430
              DATE FORMAT  YYYYXXXX   YEAR(1958)
01 BASE-DATE                                         580430
              DATE FORMAT    YYXXXX   YEAR(2058)
02 BASE-YEAR                    9(2)                 58
02 BASE-MONTH                   9(2)                 04
02 BASE-DAY                     9(2)                 30
```

Compiling for Basic Language Support

To apply Abend-AID XLS's basic language report, you match the displacement addresses and offsets that it provides to the compiler listing. Set the following options to put the necessary information into the compiler listing.

VS COBOL II

- OFFSET: Table matching statement numbers to program displacements.
- (or LIST): Expansion of COBOL code to its Assembler equivalent (contains a displacement for each Assembler instruction along with the related statement number).
- MAP: Table matching field names to offsets from BLW, BLL, etc., cells.

OS VS COBOL

- CLIST: Table matching statement numbers to program displacements.
- (or PMAP): Expansion of COBOL code to its Assembler equivalent (contains a displacement for each Assembler instruction along with the related statement number).
- DMAP: Table matching field names to offsets from BL, BLL cells.

Resolving an S0C7 with Basic Language Support

When you access a report section through the Compuware/VF, your site's System Display and Search Facility (SDSF), or printed output — without the applicable Compuware language processor in use — the report does not provide source-level support. This comprises Abend-AID XLS basic language support.

The following procedure shows how to read a basic language report on the same error and program used in "Resolving an S0C7 with Extended Language Support" on page 3-1.

1. **Review the Error Analysis section to identify the error.**

   First review the Error Analysis section, shown in Figure 3-6, for the analysis of the problem. Usually, all the information you need is in this section. Find the name of the program in error.

Figure 3-6. Error Analysis Section

```
A data exception, S0C7, occurred during execution of program PAYROLLX.

****************************************
*          Analysis of Error           *
****************************************

A Data Exception was caused by data referenced at displacement 020 from
the start of BLW cell 00 (X'0').  The field contains X'CC3311OF'.  Refer
to the data division map in the program listing to locate the field
name.

The field causing the exception is located in Working-Storage of program
PAYROLLX.

Because the program is re-entrant, Working-Storage is located outside of
the program.

--------------------------------------------------------------------------

A Data Exception is caused when a computational-3 field has
an invalid digit (not 0-9), or its last byte contains an invalid
sign (not A, B, C, D, E, or F).
```

2. **Find the name of the program in error.**

   Review the Error Location section in your Abend-AID XLS report. The Error Location section, shown in Figure 3-7 on page 3-8, identifies the program name and the displacement of the next sequential instruction to be executed.

   The Error Location section also provides the following information:

   - Program's compile date and length
   - Program's link date and load module length
   - Load module name and the load library name
   - Location of the last I/O operation or subroutine call, if applicable.

Figure 3-7. Error Location Section

```
*****************************************
*             Error Location            *
*****************************************

The next sequential instruction to be executed in program PAYROLLX was
at displacement 0000065A.

The program was compiled on 08 FEB 2002 and is 00000FB8 bytes long.

It is part of load module PAYLOAD.

The module was loaded from STEPLIB library CW.AA.TEMPLOA7.

The module was link edited on 08 FEB 2002 and is 00001400 bytes long.

The last known I/O request or call in the program above was issued with
a return address at displacement 000005AC.
```

3. **Obtain a copy of the correct compiler listing.**

   Make sure that the date of your compiler listing exactly matches the date provided by Abend-AID XLS. Look at the compile date provided in the Error Location section, as shown in Figure 3-7. Compare it to the date in the compiler listing shown in Figure 3-8. If you had multiple compiles on that date, look at the compiled program length that is also provided in the Error Location section.

Figure 3-8. Compiler Listing – Date

```
1PP 5668-958 IBM VS COBOL II Release 3.2 09/05/1990                PAYROLLX Date 02/15/2000 Time 15:12:45   Page     2
   LineID  PL SL  ----+-*A-1-B--+----2----+----3----+----4----+----5----+----6----+----7-Ż--+----8  Map and Cross Reference
0 000001               IDENTIFICATION DIVISION.
  000002               PROGRAM-ID. 'PAYROLLX' .
  000003               AUTHOR. J. PROGRAMMER.
  000004              *REMARKS.
  000005              *   THIS PROGRAM PROCESSES PAYROLL TIME-CARD RECORDS AND
  000006              *   WRITES AN EMPLOYEE MASTER RECORD FOR EACH EMPLOYEE
  000007              *   TIME-CARD. IT ALSO PRINTS A PAYROLL REPORT FOR ALL
  000008              *   EMPLOYEES.
  000009              *-----------------------------------------------------------------
  000010               ENVIRONMENT DIVISION.
  000011               CONFIGURATION SECTION.
```

4. **Find the source code of the statement in error.**

   Review the Assembler listing section of the compiler listing, shown in Figure 3-9. Find the statement number for the instruction at the displacement shown in the Error Location section. This statement is the cause of the error. The displacement given in the Error Location section is 65A. Match that displacement to the Assembler listing. In this example, the error is in statement 115. Look at the program compiler listing in Figure 3-10 on page 3-9, and match the statement number.

Figure 3-9. Assembler Listing

```
000114  MOVE
    00063A  D21F 9199 903A        MVC  409(32,9),58(9)        RPT-EMP-NAME              TCR-EMP-NAME
000115  SUBTRACT
    000640  F236 9020 905A        PACK 32(4,9),90(7,9)        WS-NET-PAY                TCR-GROSS-PAY
    000646  960F 9023              OI   35(9),X'0F'            WS-NET-PAY+3
    00064A  F235 D160 9061        PACK 352(4,13),97(6,9)      TS2=0                     TCR-TOTAL-TAXES
    000650  960F D163              OI   355(13),X'0F'          TS2=3
    000654  FB33 9020 D160        SP   32(4,9),352(4,13)      WS-NET-PAY                TS2=0
    00065A  F833 9020 9020        ZAP  32(4,9),32(4,9)        WS-NET-PAY                WS-NET-PAY
000117  MOVE
    000660  D20A D160 A0CF        MVC  352(11,13),207(10)     TS2=0                     PGMLIT AT +203
    000666  F236 D170 905A        PACK 368(4,13),90(7,9)      TS2=16                    TCR-GROSS-PAY
```

Figure 3-10. Compiler Listing – Statement in Error

```
000105                IF TCR-EMP-NO > WS-EMP-NO                                 52 50
000106      1             MOVE TCR-EMP-NO TO WS-EMP-NO                         52 50
000107                ELSE
000108      1             DISPLAY 'EMPLOYEE# SEQ ERROR  ',
000109      1                     TCR-EMP-NO,                                   52
000110      1                     SPACE,                                        IMP
000111      1             WS-EMP-NO UPON SYSOUT,                               50
000112      1             GO TO READ-A-RECORD.                                 100
000113                MOVE TCR-EMP-NO TO RPT-EMP-NO.                            52 77
000114                MOVE TCR-EMP-NAME TO RPT-EMP-NAME.                        54 75
000115                SUBTRACT TCR-TOTAL-TAXES FROM TCR-GROSS-PAY               56 55
000116                    GIVING WS-NET-PAY                                     49
000117                MOVE TCR-GROSS-PAY TO RPT-GROSS-PAY.                      55 81
000118                MOVE TCR-TOTAL-TAXES TO RPT-TOT-TAXES.                    56 83
000119                MOVE WS-NET-PAY TO RPT-NET-PAY.                           49 85
000120                ADD WS-NET-PAY TO WS-TOTAL-NET-PAY.                       49 48
```

5. Determine the field in error.

The statement in error contains three fields, as shown in Figure 3-10. Refer again to the Error Analysis section, shown in Figure 3-6 on page 3-7. Abend-AID XLS identifies the cause of the error and the field causing the data exception. The field is located at hexadecimal displacement 020 from the beginning of BLW cell 00. This field contains X'CC33110F'.

To find the field name, review the MAP/DMAP for your program, shown in Figure 3-11. The name of the field located at BLW cell 00, displacement 020, is WS-NET-PAY.

This program indicates that data is being moved into WS-NET-PAY. The invalid data is from either TCR-TOTAL-TAXES or TCR-GROSS-PAY. The MAP/DMAP shows you that TCR-TOTAL-TAXES is at displacement 61 from BLW cell 00, and TCR-GROSS-PAY is at displacement 5A. Review the Working-Storage section for your program in the Abend-AID XLS report, shown in Figure 3-12 on page 3-10. The contents of working storage are grouped by BLW cell. Abend-AID XLS working storage indicates that TCR-TOTAL-TAXES contains 013222, and TCR-GROSS-PAY contains @@33110, which is invalid data.

Figure 3-11. MAP/DMAP

```
Source   Hierarchy and                          Base       Hex-Displacement  Asmblr Data                  Data Def
LineID   Data Name                             Locator    Blk   Structure   Definition    Data Type      Attributes
     2  PROGRAM-ID PAYROLLX ----------------------------------------------------------------------------------*
    23   FD MASTER. . . . . . . . . . . . . .                                            QSAM           FB
    29   01 MASTER-RECORD . . . . . . . . . . BLF=0000   000               DS 80C        Display
    30   FD OUTLIST . . . . . . . . . . . . .                                            QSAM           FB
    36   01 DATA-LINE . . . . . . . . . . . . BLF=0001   000               DS 133C       Display
    37   FD TIMECARD. . . . . . . . . . . . .                                            QSAM           FB
    43   01 TIMECARD-RECORD . . . . . . . . . BLF=0002   000               DS 80C        Display
    45   77 WS-LINE-CTR . . . . . . . . . . . BLW=0000   000               DS 2P         Packed-Dec
    46   77 WS-PAGE-CTR . . . . . . . . . . . BLW=0000   008               DS 2P         Packed-Dec
    47   77 WS-RECORD-CTR . . . . . . . . . . BLW=0000   010               DS2P          Packed-Dec
    48   77 WS-TOTAL-NET-PAY. . . . . . . . . BLW=0000   018               DS 5P         Packed-Dec
    49   77 WS-NET-PAY. . . . . . . . . . . . BLW=0000   020               DS 4P         Packed-Dec
    50   77 WS-EMP-NO . . . . . . . . . . . . BLW=0000   028               DS 8C         Disp-Num
    51   01 WS-TIMECARD-REC . . . . . . . . . BLW=0000   030               DS 0CL80      Group
    52     02 TCR-EMP-NO. . . . . . . . . . . BLW=0000   030   0 000 000   DS 8C         Disp-Num
    53     02 TCR-MONTH-CODE. . . . . . . . . BLW=0000   038   0 000 008   DS 2C         Disp-Num
    54     02 TCR-EMP-NAME. . . . . . . . . . BLW=0000   03A   0 000 00A   DS 32C        Display
    55     02 TCR-GROSS-PAY . . . . . . . . . BLW=0000   05A   0 000 02A   DS 7C         Disp-Num
    56     02 TCR-TOTAL-TAXES . . . . . . . . BLW=0000   061   0 000 031   DS 6C         Disp-Num
    57     02 TCR-FILLER. . . . . . . . . . . BLW=0000   067   0 000 037   DS 25C        Display
```

Figure 3-12. Working-Storage Section

```
              Working-Storage Section

Working storage referenced by BLW cell      0 (X'0')

 DSPL Address   -------------- Data ---------------
00000 06902650  004C0000 00000000 001C0000 00000000  .<..............
00010 06902660  004C0000 00000000 00134042 3C000000  .<........ .....
00020 06902670  CC3311OF 00000000 F0F4F0F6 F0F2F9F3  ........04060293
00030 06902680  F0F4F0F6 F0F2F9F3 F0F2C5C4 E6C1D9C4  0406029302EDWARD
00040 06902690  E26B40C1 D3C5E7C1 D5C4C5D9 40404040  S, ALEXANDER
00050 069026A0  40404040 40404040 40407C7C F3F3F1F1            @@3311
00060 069026B0  F0F0F1F3 F2F2F2F4 F3F2F3F4 F3F2F440  0013222432343244

                              Linkage Section
Linkage storage referenced by BLL cell      1 (X'1')
 (Length:  EXEC PARM)

 DSPL Address   -------------- Data ---------------
00000 00005FEE  00074040 40404040 40                ..

                              Supporting data

                           Contents of BLW cells
   BLW      0 (X'0')    - 06902650

                           Contents of BLL cells
   BLL      0 (X'0')    - 00000000   BLL      1 (X'1')    - 00005FEE

                            Task Global Table (TGT)
 DSPL Address   -------------- Data ---------------
00000 06902028  00108001 00005F98 000487B0 869013A2  ......¬q..g.f..s
00010 06902038  80019178 86901E0E 00049FE0 000052D4  ..j.f......\...M
00020 06902048  06902200 06902318 869011AC 0004D220  ........f.....K.
00030 06902058  00049FE0 069025B0 06902650 06900CAC  ...\.......&....
```

6. **Find the contents of other data fields in your program.**

   Use Abend-AID XLS to investigate the contents of other data fields in your program. In this example, find the contents of WS-RECORD-CTR. Refer again to the Working-Storage section shown in Figure 3-12. Use the following procedure to find the contents of WS-RECORD-CTR.

   a. Find the BLW cell and displacement for WS-RECORD-CTR in the program's MAP/DMAP.
   b. Look at the Working Storage portion for BLW cell 00 and "match" the displacement of 010 using the DSPL column.

   In this example, the match indicates that WS-RECORD-CTR was set at 4 (X'004C').

   To find the contents of an indexed field, refer to "Determining Index and Indexed Field Values" on page 3-11.

   Review record information. The File section shows information about all files that are open at the time of the error. This information includes identification of the current record and previous records, when available. DCB and other control block information is also given. Abend-AID XLS displays records using decimal locations, making fields easier to find when using record layouts. File information is provided for all file types, including VSAM and IAM.

   You can directly access these files through File-AID from the online File Summary Selection List, which lists multiple open files. Inside a file you have full File-AID edit and browse capabilities.

Searching Program Storage

The commands shown below enable you to immediately access the specific part of program storage that you need, if you're working with basic language support. Enter the commands within the Program Storage section of the Abend-AID XLS report.

PROG(progname) BL x

Locates the cell specified by *x* in the program specified by *progname*. Specify **x'x'** for hexadecimal cell numbers.

BL x, BLL x, BLW x, BLX x

Locates the base locator cell specified by *x* in the current program storage.

PROG(progname) TGT

Locates the TGT in the program specified by *progname*.

TGT

Locates the TGT in the current program.

EXT

Displays the COBOL External Data Section.

Determining Index and Indexed Field Values

The following procedures tell how to determine an indexed occurrence within a table using a basic language report.

VS COBOL II, COBOL/370, and COBOL for MVS and VM

1. Find the Program Storage section for your program in the Abend-AID XLS report. At the end of the section, find "Contents of index cells" under "Supporting data," as shown in Figure 3-13. Note that the index cells are listed in the same order as they appear in the Working-Storage section in your program. Locate the index cell that corresponds to the desired index.

Figure 3-13. Supporting Data Section

```
                          Supporting data

                        Contents of BLW cells

BLW       0 (X'0')    - 00013CA0

                        Contents of BLL cells

BLL       0 (X'0')    - 00000000   BLL       1 (X'1')    - 00000000
BLL       2 (X'2')    - 00000000

                 No BLF cells used in this program

                        Contents of index cells

IDX       0 (X'0')    - 0000001B   IDX       1 (X'1')    - 00000024
```

2. To calculate the index value or occurrence, do the following:

   a. Convert the value of these four bytes from hexadecimal into decimal. For example, if the four bytes contain X'0000001B', convert hexadecimal 1B to decimal 27.

b. Divide this decimal value by the decimal length of one occurrence within the table. For example, if the length of one occurrence is 9 bytes and the index value in decimal is 27: 27 / 9 = 3.
c. Add one to this value: 3 + 1 = 4.

The result is the occurrence within the table that the index was pointing to when the Abend-AID XLS report was created.

Note: The index cells are based from the TGT in COBOL II or COBOL/370, and from the TGT or DSA in COBOL for MVS and VM. Supporting data will identify the TGT or DSA index cells.

OS/VS COBOL

1. In the Memory Map at the end of your program's compiler listing, find the entry for "INDEX CELLS," as in Figure 3-14. Make note of the displacement.

Figure 3-14. Memory Map

```
                        MEMORY MAP

         TGT                              00118

  SAVE AREA                               00118
  SWITCH                                  00160
  TALLY                                   00164
  .
  .
  .
  BLL CELLS                               00310
  VLC CELLS                               00344
  SBL CELLS                               00344
  INDEX CELLS                             00344
  SUBADR CELLS                            00348
```

2. Find the Program Storage section for your program in the Abend-AID XLS report. In the task global table (TGT), locate the displacement specified for "INDEX CELLS" in the Memory Map, as shown in Figure 3-15.

Figure 3-15. Task Global Table (TGT)

```
                          Task Global Table (TGT)

  DSPL  Address   -------------- Data ---------------
  00118 00008348  00300000 000073C8 00000000 00000000  .......H........
  00128 00008358  00000000 00000000 00000000 00000000  ................
  .
  .
  .
  002C8 000084F8  00008230 00000000 00007BE0 E2E8E2D6  ..b.......#\SYSO
  002D8 00008508  E4E34040 E380000C 000082B8 00000000  UT  T.....b.....
  002E8 00008518  00000000 00000000 00000000 00000000  ................
  002F8 00008528  00008590 00000000 00000000 00000000  ..e.............
  00308 00008538  00000000 00000000 00000000 00000000  ................
  00318 00008548  000082D0 00000000 00000000 0000012F  ..b}............
  00328 00008558  00000000 0000000C 00000000 00000000  ................
  00338 00008568  00006D78 80006F0D 00000000 0000001B  ..._...?........
```

3. At the located TGT displacement, note that the index contents are listed in the same order as they appear in the Working-Storage section in your program. Locate the index contents hat correspond to the desired index. The contents of each index cell are four bytes in length.

4. To calculate the index value, do the following:

   a. Convert the value of these four bytes from hexadecimal into decimal. For example, if the four bytes contain X'0000001B', convert hexadecimal 1B to decimal 27.
   b. Divide this decimal value by the decimal length of one occurrence within the table. For example, if the length of one occurrence is 9 bytes and the index value in decimal is 27: 27 / 9 = 3.
   c. Add one to this value: 3 + 1 = 4.

   The result is the occurrence within the table that the index was pointing to when the Abend-AID XLS report was created.

Chapter 4.
Diagnosing a PL/I Data-Related Error

This chapter describes how to use Abend-AID XLS to diagnose a common PL/I error. Abend-AID XLS provides two levels of PL/I support: extended language support and basic language support.

Some conditions or characteristics found in PL/I programs restrict the usage of PL/I XLS. The usage restrictions are listed in Chapter 8 of the Compuware Shared Services User/Reference Guide.

Resolving an On-Code 0320 with Extended Language Support

When you select a report section online through the Compuware/VF Viewing Facility (Compuware/VF), with the applicable Compuware language processor in use, program source code can be automatically merged into the report. This comprises Abend-AID XLS (Extended Language Support).

PL/I XLS provides:

- The source code of the statement in error displayed in the Error Analysis section. For data-related errors, all PL/I variables in the statement in error are shown with their current values.
- The source code of the statement being executed in the Error Location section.
- The names and contents of all individual PL/I variables in storage displayed in the Program Storage section.
- The source listings for PL/I programs on the calling chain displayed in the Program Listing section with the current statement indicated.

The following procedure shows how to read an extended language report on a common error, an on-code 0320 abend.

1. Review the Error Analysis section to determine the cause of the error.

   First review the Error Analysis section, shown in Figure 4-1, to find the cause of the error. Usually, all the information you need is in this section. The source code of the statement in error is displayed, and all variables are shown with the current values. The Error Analysis section also gives a description of an on-code 0320. This example is a decimal divide exception, which is caused when the divisor is zero. The divisor in the statement in error is MONTH_CODE.

Figure 4-1. Error Analysis Section

```
   The user abend code of U3001 was issued by PL/I after normal return from
   an error or finish on-unit for error on-code 0320.  That on-code
   corresponds to the OS completion code of S0CB.

   ****************************************
   *          Analysis of Error           *
   ****************************************

   A Decimal Divide Exception occurred in the following statement:

000065            AVG_HOURS = TOTAL_HOURS / MONTH_CODE;

              Current values of fields on abending statement
--- Variable/Attributes -----------    Value----1----+----2----+----3----+----4

CALC_PAYROLL:AVG_HOURS
    AUTO    FIXED DEC(5,1)             +43.0
CALC_HOURS:TOTAL_HOURS
    STATIC  FIXED DEC(5)               +101
PAYROLL:TIME_RECORD.MONTH_CODE
    AUTO    PIC'99'                    00
-------------------------------------------------------------------------------
               "#" - Indicates field contains invalid data

  A decimal divide exception is caused when the divisor is zero.

                          Specific Information
         CSECT    DISPL/   Module   DISPL/   DISPL/
  ADDR   Name     CSECT    Name      EPA      Load    Len Instruction
06900F2E PAYROLL1 000009DE PAYEXEC  00000A5E 00000A5E  6  FD91 D098 D0C8
                                                        DP - divide decimal

                                                     A-operand - dividend

00011F18                                              10  0000001010000000000000C

                                                     B-operand - divisor

00011F48                                               2  000C
                                                  The above field is in error
```

2. Review the Error Location section to find the location of the error.

   The Error Location section, shown in Figure 4-2, indicates that the statement being executed is contained in procedure CALC_HOURS.

   The Error Location section also provides the following information:

   - Program's compile date and length
   - Program's link date and load module length
   - Load module name and the load library name
   - Location of the last I/O operation or subroutine call, if applicable.

Figure 4-2. Error Location Section

```
   ****************************************
   *             Error Location           *
   ****************************************

   The statement being executed was:

000065             AVG_HOURS = TOTAL_HOURS / MONTH_CODE;

   This statement is contained in
   procedure PAYROLL:CALC_PAYROLL:CALC_HOURS.

   The program was compiled on 08 FEB 2002 and is 00000A28 bytes long.

   It is part of load module PAYEXEC.

   The module was loaded from STEPLIB library CW.AA.TEMPLOA7.

   The module was link edited on 08 FEB 2002 and is 00003B30 bytes long.

   The last known I/O request or call in the program above was:

000063              OVERTIME_TOTAL.ACCRUED_HOURS = OVERTIME_TOTAL.ACCRUED_HOURS +
                        OVERTIME_HOURS_WEEKLY;

   This statement is contained in
   procedure PAYROLL:CALC_PAYROLL:CALC_HOURS.
```

The Error Analysis and Error Location sections usually provide enough information to resolve the problem. The other sections of the Abend-AID XLS report provide supporting information that can be used to trace the cause of the error.

3. Find the contents of other variables in your program.

   To find the contents of other variables, review the Program Storage section in the Abend-AID XLS report. Program storage is formatted by individual variable name in the same order as declared in the program. Variable name, storage type, definition, and current value for each PL/I variable located in storage at the time of the error are given, as shown in Figure 4-3 on page 4-4 through Figure 4-5 on page 4-5.

Figure 4-3. Program Storage Section – Procedure PAYROLL

```
                Variable Storage - Procedure PAYROLL
--- Variable/Attributes -----------    Value----1----+----2----+----3----+----4


1   TIME_RECORD
    AUTO
 2  EMPLOYEE_NUMBER_REMOTE_LOCATION
            CHAR(8)                    30900200
 2  MONTH_CODE
            PIC'99'                    00
 2  EMP_NAME
            CHAR(30)                   SMITH,  ROBERT  M.
 2  GROSS_PAY
            CHAR(7)                    0232469
 2  TOTAL_TAXES
            CHAR(5)                    01230
 2  SALARY_PER_HOUR
            CHAR(4)                    3200
 2  REGULAR_HOURS_WEEKLY
            CHAR(3)                    018
 2  OVERTIME_HOURS_WEEKLY
            CHAR(3)                    040
 2  COMMENT
            CHAR(18)                   03PNC 0320 SAMPLE
.
.
1   REGULAR_TOTAL
    AUTO
 2  EMP_NUMBER
            CHAR(8)                    0
 2  ACCRUED_HOURS
            FIXED DEC(3,1)             +53.0
SORT_EMPL
    AUTO    CHAR(17)                   DECEMBER 40140039
NET_PAY
    AUTO    FIXED DEC(7,2)             +67999.00
MONTHS
            EXT  ENTRY
ADDR
    BUILTIN
```

Figure 4-4. Program Storage Section – Procedure CALC_HOURS

```
                Variable Storage - Procedure CALC_HOURS      ---
Variable/Attributes -----------    Value----1----+----2----+----3----+----4

    TOTAL_HOURS
        STATIC  FIXED DEC(5)              +101
```

Figure 4-5. Program Storage Section – Procedure CALC_PAYROLL

```
                    Variable Storage - Procedure CALC_PAYROLL
--- Variable/Attributes -----------    Value----1----+----2----+----3----+----4

 AVG_HOURS
     AUTO    FIXED DEC(5,1)             +43.0

 1   CNTL_STRUCT                (0)
     CTL
  2  TABLE_NAME
             CHAR(30)                   JOHNSTONE, REGINALD B.
  2  TABLE_DATA (1:2)
             (1)  FLT DEC(5)            +2.000000E+0
             (2)  FLT DEC(5)            +2.000000E+0
  2  TABLE_ACTIVITY
             CHAR(18)                   02RECORD 2

 1   CNTL_STRUCT               (-1)
     CTL
  2  TABLE_NAME
             CHAR(30)                   JONES,  KERI, M.
  2  TABLE_DATA (1:2)
             (1)  FLT DEC(5)            +1.000000E+0
             (2)  FLT DEC(5)            +1.000000E+0
  2  TABLE_ACTIVITY
             CHAR(18)                   01RECORD 1

 1   INVENTORY_RANGE (1:3)
     AUTO
  2  (1)  WAREHOUSE
  3  MAX_STOCK
             (1)  FIXED DEC(4,1)        +99.0
  3  MIN_STOCK
             (1)  FIXED DEC(3,1)        +88.0
  2  (1)  ROW_BIN
  3  MAX_STOCK
             (1)                        +99.0
  3  MIN_STOCK
             (1)                        +88.0
 .
  2  (3)  WAREHOUSE
  3  MAX_STOCK
             (3)  FIXED DEC(4,1)        +0.0
  3  MIN_STOCK
             (3)  FIXED DEC(3,1)        +0.0
  2  (3)  ROW_BIN
  3  MAX_STOCK
             (3)                        +0.0

 --- Variable/Attributes -----------    Value----1----+----2----+----3----+----4

  3  MIN_STOCK
             (3)                        +0.0

 WAREHOUSE_DESCRIPTION
     AUTO    CHAR(120)                  DESCRIPTION DESCRIPTION DESCRIPTION DESC
                                +40     RIPTION DESCRIPTION DESCRIPTION DESCRIPT
                                +80     ION DESCRIPTION DESCRIPTION DESCRIPTION
 WAGES (-2:1,1:2)
     AUTO
          (-2,1)  FIXED DEC(3)          +2
          (-2,2)  FIXED DEC(3)          +2
          (-1,1)  FIXED DEC(3)          +0
          (-1,2)  FIXED DEC(3)          +0
           (0,1)  FIXED DEC(3)          +0
           (0,2)  FIXED DEC(3)          +0
           (1,1)  FIXED DEC(3)          +0
           (1,2)  FIXED DEC(3)          +0

 1   SECTION_FLAGS
     STATIC
  2  SECTION1_FLG
             BIT(2)                  B'10'
  2  SECTION2_FLG
             BIT(2,2)                B'00'
  2  SECTION3_FLG
             BIT(2,4)                B'10'
  2  SECTION4_FLG
             BIT(2,6)                B'00'
```

4. Review other supporting information.

   Whenever PL/I source listings are available in the source listing dataset, XLS provides the source listings for all active PL/I programs with the current statement indicated. This information is shown in the Program Listing section. Figure 4-6 shows the program listing for PAYROLL.

Figure 4-6. Program Listing Section

```
      ************************************************************************
      *              Program Listing Section - PAYROLL                       *
      ************************************************************************

000059        CALC_HOURS: PROC;
000060        DCL     TOTAL_HOURS STATIC  FIXED DEC(5);
000061            REGULAR_TOTAL.EMP_NUMBER = OVERTIME_TOTAL.EMP_NUMBER =
                      EMPLOYEE_NUMBER_REMOTE_LOCATION;
000062            REGULAR_TOTAL.ACCRUED_HOURS = REGULAR_TOTAL.ACCRUED_HOURS +
                      REGULAR_HOURS_WEEKLY;
000063            OVERTIME_TOTAL.ACCRUED_HOURS = OVERTIME_TOTAL.ACCRUED_HOURS +
                      OVERTIME_HOURS_WEEKLY;
000064            TOTAL_HOURS = OVERTIME_TOTAL.ACCRUED_HOURS +
                                REGULAR_TOTAL.ACCRUED_HOURS;
 CURR STMT ==>    AVG_HOURS =  TOTAL_HOURS / MONTH_CODE;
000066            SORT_EMPL = MONTH žž EMPLOYEE_NUMBER_REMOTE_LOCATION;
000067         END CALC_HOURS;
```

5. Review record information.

   The File section shows information about all files that are open at the time of the error. This information includes identification of the current record and previous records, when available. DCB and other control block information is also given. Abend-AID XLS displays records using decimal locations, making fields easier to find when using record layouts. File information is provided for all file types, including VSAM and IAM.

   You can directly access these files through File-AID from the online File Summary Selection List, which lists multiple open files. Inside a file you have full File-AID edit and browse capabilities.

Searching Program Listing

The commands shown below enable you to immediately access the source statements that you need, if you're working with XLS. Enter them as shown in an open Program Listing section with the PLIST command.

blank

Displays source statements as determined by the SLSLIST or SLSMLIST option in the CWGLOBAL table.

nn

Displays *nn* lines before and after the current statement.

ALL

Displays the complete PL/I source listing.

LABEL

Displays the label portion of the PL/I program containing the current statement.

Types of Basic Language Support

Abend-AID provides non-abending and abending support for PL/I. The type of support your site installs depends on how your PL/I application programs presently terminate following an error.

- **Non-abending support:** Diagnoses the error and returns control to PL/I. This allows the program to terminate with the appropriate PL/I return code.
- **Abending Support:** Issues a U3001 abend. Abend-AID is then invoked through normal SVC51 processing to diagnose the error. Abending support lets your site avoid coding an IBMBEER or IBMBXITA exit in order to abend in an IMS environment.

  Note: The Abend-AID SVC51 interface must be started when using either non-abending or abending PL/I support.

When PL/I is used with Language Environment (LE), Abend-AID receives control via LE condition-handling rather than PL/I condition-handling. Therefore, if you are using PL/I exclusively with LE, Compuware recommends that your site install Abend-AID Language Environment support instead of Abend-AID PL/I support.

Resolving an On-Code 0320 with Basic Language Support

When you access a report section through the Compuware/VF, your site's System Display and Search Facility (SDSF), or printed output — without the applicable Compuware language processor in use — the report does not provide source-level support. This comprises Abend-AID XLS basic language support.

PL/I basic support provides:

- Diagnostics for all PL/I errors including full ON-UNIT support.
- The procedure name and the displacement of the statement in error. The statement number is given when the GOSTMT compiler option is used.
- The line number, when the GONUMBER compiler option is used.
- A call trace showing procedure names and displacements. Statement numbers are shown when the GOSTMT compiler option is used.
- Automatic, static, and other PL/I storage areas. In the automatic storage area, Abend-AID XLS indicates the offset to the main entry point at the time of the PROC block, ON-UNIT, or library procedure transferred control with a CALL or BEGIN statement. Abend-AID XLS shows both the absolute addresses and their corresponding relative addresses, which match the offsets of the static variables on the variable storage map. Thus, you can directly reference each static variable. The names of the PROC, CSECT, and load module are provided.

The following procedure shows how to read a basic language report on the same error and program that was used in "Resolving an On-Code 0320 with Extended Language Support" on page 4-1.

1. Review the Error Analysis section to identify the error.

   First review the Error Analysis section, shown in Figure 4-8 on page 4-8, for the analysis of the problem. Usually, all the information you need is in this section. The Error Analysis section also gives a definition of an on-code 0320. This example is a decimal divide exception, which is caused when the divisor is zero.

Figure 4-7. Error Analysis Section

```
   The user abend code of U3001 was issued by PL/I after normal return from
   an error or finish on-unit for error on-code 0320.  That on-code
   corresponds to the OS completion code of S0CB.

   ******************************************
   *          Analysis of Error             *
   ******************************************

   A decimal divide exception is caused when the divisor is zero.

                           Specific Information
          CSECT    DISPL/   Module   DISPL/   DISPL/
   ADDR   Name     CSECT    Name      EPA      Load    Len Instruction
06900F2E PAYROLL1 000009DE PAYEXEC  00000A5E 00000A5E  6  FD91 D098 D0C8
                                                        DP - divide decimal

                                                     A-operand - dividend

00011F18                                              10  0000001010000000000000C

                                                     B-operand - divisor

00011F48                                                2  000C
                                                  The above field is in error

Note: The CSECT and module displacements are not applicable
      because the operands are located in storage outside
      of the program.
```

2. Find the name of the procedure in error.

   Review the Error Location section in your Abend-AID XLS report. The Error Location section, shown in Figure 4-8, identifies the procedure name and the offset of the next sequential instruction to be executed. The GOSTMT compiler option was used to get the statement number.

   The Error Location section also provides the following information:

- Program's compile date and length
  - Program's link date and load module length
  - Load module name and the load library name
  - Location of the last I/O operation or subroutine call, if applicable.

Figure 4-8. Error Location Section

```
  ******************************************
  *             Error Location             *
  ******************************************

  The next sequential instruction to be executed was in statement 65 at
  offset 00000016C in procedure CALC_HOURS.

  The program was compiled on 08 FEB 2002 and is 00000A28 bytes long.

  It is part of load module PAYEXEC.

  The module was loaded from STEPLIB library CW.AA.TEMPLOA7.

  The module was link edited on 08 FEB 2002 and is 00003B30 bytes long.

  The last known I/O request or call in the program above was issued with
  a return address at displacement 0000097C.
```

3. Obtain a copy of the correct compiler listing.

   Make sure that your compiler listing matches exactly with the date provided by Abend-AID XLS. Look at the compile date provided in the Error Location section in

Figure 4-8 on page 4-8. Compare it to the date in the compiler listing shown in Figure 4-9. If you had multiple compiles on that date, then look at the compiled program length that is also provided in the Error Location section.

Figure 4-9. Compiler Listing – Date

```
15668-
910 IBM OS PL/I OPTIMIZING COMPILER  VER 2 REL 3 MOD 0       05 MAR 2000  12:30:39  PAGE  1
-OPTIONS SPECIFIED
OAG,LIST,MAP,OFFSET,STORAGE,A(F),XREF(F),OPT,GOSTMT;
.
.
.
15668-
910 IBM OS PL/I OPTIMIZING COMPILER   PAYROLL: PROC OPTIONS (MAIN);                PAGE  2
-                  SOURCE LISTING
-    STMT
0
        1   PAYROLL: PROC OPTIONS (MAIN);
               /*************************************************************/
               /* THIS PROGRAM PROCESSES PAYROLL TIME-CARD RECORDS AND */
```

4. Look at the source code of the statement in error.

   The Error Location section, shown in Figure 4-8 on page 4-8, provides the number for the statement in error. Look at the program compiler listing in Figure 4-10, and match the statement number.

Figure 4-10. Compiler Listing – Statement in Error

```
    63      OVERTIME_TOTAL.ACCRUED_HOURS = OVERTIME_TOTAL.ACCRUED_HOURS +
                OVERTIME_HOURS_WEEKLY;
    64      TOTAL_HOURS = OVERTIME_TOTAL.ACCRUED_HOURS +
                          REGULAR_TOTAL.ACCRUED_HOURS;

    65      AVG_HOURS =  TOTAL_HOURS / MONTH_CODE;

    66      SORT_EMPL = MONTH ŽŽ EMPLOYEE_NUMBER_REMOTE_LOCATION;
    67   END CALC_HOURS;
    68   END CALC_PAYROLL;
    69   WRITE_TOTAL:
             TOTAL_NET_PAY  = TOTAL_CTR;
    70       WRITE FILE (OUTLIST) FROM (FINAL_LINE);
```

5. Determine the variable in error.

   The statement in error contains three variables, as shown in Figure 4-10. Refer again to the Error Analysis section, shown in Figure 4-8 on page 4-8. Abend-AID XLS identifies the cause of the error and the variable causing the decimal divide exception. The Specific Information area of the Error Analysis section indicates that the B-operand is the field in error. This variable contains X'000C'. The divisor in the statement in error is MONTH_CODE.

6. Find the contents of other variables in your program.

   Use Abend-AID XLS to investigate the contents of other variables in your program. In this example, find the contents of RECORD_CTR. Refer to the variable storage map shown in Figure 4-11 on page 4-10. Find the storage CLASS and HEX offset for RECORD_CTR.

Figure 4-11. Variable Storage Map

```
-                              VARIABLE STORAGE MAP -
IDENTIFIER                    LEVEL         OFFSET       (HEX)   CLASS     BLOCK

 DYNAMIC_LABEL                    1            272         110   AUTO      PAYROLL
 SORT_EMPL                        1            296         128   AUTO      PAYROLL
 NET_PAY                          1            284         11C   AUTO      PAYROLL
 RETCODE                          1            280         118   AUTO      PAYROLL
 EOF                              1            288         120   AUTO      PAYROLL
 REGULAR_TOTAL                    1            313         139   AUTO      PAYROLL
 EMP_NUMBER                       1            313         139   AUTO      PAYROLL
 ACCRUED_HOURS                    1            321         141   AUTO      PAYROLL
 COUNTERS                         1           1072         430   STATIC    PAYROLL

 RECORD_CTR                       1           1072         430   STATIC    PAYROLL

 PAGE_CTR                         1           1074         432   STATIC    PAYROLL
 TOTAL_CTR                        1           1076         434   STATIC    PAYROLL
 LINE_CTR                         1           1081         439   STATIC    PAYROLL
```

Note: If you can't locate a variable in the variable storage map, refer to the compiler listing, shown in Figure 4-12, to check the DECLARE statement. If it has an attribute of EXTERNAL, the statement is defined outside of the current program and has an absolute address that is located outside of the current program's automatic or static storage areas.

Figure 4-12. Compiler Listing

```
 9       DECLARE
             SORT_EMPL              CHAR(17),
             NET_PAY                FIXED(7,2),
             RETCODE                FIXED BIN(31,0),
             EOF                    BIT(1) INIT('0'B);
10  DCL  DYNAMIC_LABEL    LABEL;
11  DCL  MONTHS EXTERNAL RETURNS (CHAR(9));
        /*********************************************************/
        /*            FILE    OPENS                              */
```

With the storage class and hexadecimal offset, use the following procedure to find the contents of RECORD_CTR.

a. Look at the storage class and procedure block information provided by the PL/I variable storage map for the variable. The class identifies the appropriate storage area to use: automatic or static. The class for RECORD_CTR is STATIC, and the hexadecimal offset is 430.

b. Review the Program Storage section of the Abend-AID XLS report. In the static storage area, locate the contents of RECORD_CTR. Use the hexadecimal offset from the variable storage map and the DSPL column in the static storage area to identify the beginning of the variable, as shown in Figure 4-13 on page 4-11. In this example, the match indicates that RECORD_CTR was set at 2 (X'002C').

Figure 4-13. Program Storage Section

```
            ****************************************************
            Static storage for PL/I procedure PAYROLL
            in CSECT PAYROLL1 of load module PAYEXEC
            Compiled with options:  SYSTEM( MVS )
            ****************************************************

 DSPL Address   -------------- Data ----------------
00000 06900F78  E00005B4 06900558 0690060C 0690075C  \.............*
00010 06900F88  06900820 06900882 069008AC 0690090E  .......b........
.
.
00410 06901388  06900A58 0690090E 06900A82 0690090E  ...........b....
00420 06901398  06900ADA 0690090E 06900B04 0690090E  ................

00430 069013A8  002C001C 01000040 0C003C88 00000000  ....... ...h....
00440 069013B8  00101C00 00000000 00000001 06900558  ................
00450 069013C8  000002C6 06901440 00000001 01140004  ...F... ........
```

7. Review record information.

   The File section shows information about all files that are open at the time of the error. This information includes identification of the current record and previous records, when available. DCB and other control block information is also given. Abend-AID XLS displays records using decimal locations, making fields easier to find when using record layouts. File information is provided for all file types, including VSAM and IAM.

   You can directly access these files through File-AID from the online File Summary Selection List, which lists multiple open files. Inside a file you have full File-AID edit and browse capabilities.

Searching Program Storage

The commands shown below enable you to immediately access the specific part of program storage that you need, if you're working with basic language support. Enter the commands within the Program Storage section of the Abend-AID XLS report.

AUTO

Locates automatic storage.

STAT

Locates static storage.

CTL

Locates controlled storage.

Chapter 5.
Diagnosing an Assembler Data-Related Error

This chapter describes how to use Abend-AID XLS to diagnose a common Assembler error. Abend-AID XLS provides two levels of Assembler support: extended language support (XLS) and basic language support. Assembler XLS is applicable to High-level Assembler only.

Resolving an S0CB with Extended Language Support

When you select a report section online through the Compuware/VF Viewing Facility (Compuware/VF), with the applicable Compuware language processor in use, program source code can be automatically merged into the report. This comprises Abend-AID XLS (Extended Language Support).

Assembler XLS provides:

- The source code of the statement in error displayed in the Error Analysis section. For data-related errors, all Assembler variables in the statement in error are shown with their current values.
- The source code of the statement being executed in the Error Location section.
- The names and contents of all individual Assembler variables in storage displayed in the Program Storage section.
- The source listings for Assembler programs on the calling chain displayed in the Program Listing section with the current statement indicated.
- The following procedure shows how to read an extended language report on a common error, an S0CB abend.

1. Review the Error Analysis section to determine the cause of the error.

   First review the Error Analysis section, shown in Figure 5-1 to find the cause of the error. Usually, all the information you need is in this section. The source code of the instruction in error is displayed, and operand fields are shown with the current values. The Error Analysis section also gives a description of an S0CB abend. This example is a decimal divide exception, which is caused when the divisor is zero. The divisor in the instruction in error is XYEARS.

Figure 5-1. Error Analysis Section

```
   A decimal divide exception, SOCB, occurred during execution of program
   CWAADATE.

   ****************************************
   *          Analysis of Error           *
   ****************************************

   A Decimal Divide Exception occurred in the following statement:

    Stmt   Loc  Object Code       Source Statement
      48                        * Uncomment following line before rolling
      49                        * this program into production!
      50                        *        ZAP   XYEARS,=P'3'
      51 00003C F221 B165 6000           PACK  PACKED_YEAR,RUN_YEAR
CURR => 000042 FD20 B165 B16A            DP    PACKED_YEAR,XYEARS
      53 000048 F900 B167 B176           CP    PACKED_YEAR+2(1),=P'0'
      54 00004E 4770 B064                BNE   EXIT
      55 000052 D501 7002 6002           CLC   HIRE_MONTH,RUN_MONTH
      56 000058 4770 B064                BNE   EXIT

       Operands         Mapped-By     Def    Base Off Len           Hex
---------------------- ------------ -------- ---- --- ---  ------------------
PACKED_YEAR            CWAADATE     PL3      R11  165   3  00019F
XYEARS                 CWAADATE     P        R11  16A   1  0C
    ---------------------------------------------------------------------------

  A decimal divide exception is caused when the divisor is zero.

                           Specific Information
                          Load
Absolute Program  DISPL/   Module   DISPL/   DISPL/        Machine
Address  Name     Prog     Name     EPA      Load      Len Instruction

0AE14AA2 CWAADATE 00000042 CWAADATE 00000042 00000042   6   FD20 B165 B16A
                                                         DP - divide decimal

                                                        A-operand - dividend

0AE14BC5 CWAADATE 00000165 CWAADATE 00000165 00000165   3   00019F

                                                        B-operand - divisor

0AE14BCA CWAADATE 0000016A CWAADATE 0000016A 0000016A   1   0C
                                              The above field is in error
```

2. Review the Error Location section to find the location of the error.

   The Error Location section, shown in Figure 5-2, indicates that the instruction being executed is contained in program CWAADATE.

   The Error Location section also provides the following information:

   - Program's compile date and length
   - Program's link date and load module length
   - Load module name and the load library name
   - Location of the last I/O operation or subroutine call, if applicable.

Figure 5-2. Error Location Section

```
   ****************************************
   *             Error Location           *
   ****************************************

   The statement being executed in CSECT CWAADATE was at displacement
   00000042.

    Stmt   Loc  Object Code       Source Statement
      48                        * Uncomment following line before rolling
      49                        * this program into production!
      50                        *        ZAP   XYEARS,=P'3'
      51 00003C F221 B165 6000           PACK  PACKED_YEAR,RUN_YEAR
CURR => 000042 FD20 B165 B16A            DP    PACKED_YEAR,XYEARS
      53 000048 F900 B167 B176           CP    PACKED_YEAR+2(1),=P'0'
      54 00004E 4770 B064                BNE   EXIT
      55 000052 D501 7002 6002           CLC   HIRE_MONTH,RUN_MONTH
      56 000058 4770 B064                BNE   EXIT

   The program was compiled on 08 FEB 2002 and is 00000178 bytes long.

   It is part of load module CWAADATE.

   The module was loaded from STEPLIB library CW.AA.TEMPLOA7.

   The module was link edited on 08 FEB 2002 and is 00000178 bytes long.
```

The Error Analysis and Error Location sections usually provide enough information to resolve the problem. The other sections of the Abend-AID XLS report provide supporting information that can be used to isolate the cause of the error.

3. Find the contents of other variables in your program.

   To find the contents of other variables, review the Program Storage section in the Abend-AID XLS report. Program storage is formatted by individual field name in the same order as defined in the program. The name, type, and current value for each field mapped by an active USING instruction at the time of the error are given, as shown in Figure 5-3 on page 5-4.

Figure 5-3. Program Storage Section – CSECT CWAADATE

```
 STORAGE FOR CSECT CWAADATE (R11)
Offset Address      Line  Label    Type        ----- HEX ------ - CHAR -
------ -------- --------- -------- --------    ---------------- --------
000000 0AE14A60       6             DS 0H
0000EC 0AE14B4C     125    SAVEAREA DS 18F      0000000000030018 ........
0000F4 0AE14B54                                 0000000000000000 ........
                    132    DATE_DAYS
000134 0AE14B94                      DS 0C
000134 0AE14B94     133              DC C        F0F1F3F1         0131
000138 0AE14B98     134              DC C        F0F2F2F8         0228
00013C 0AE14B9C     135              DC C        F0F3F3F1         0331
000140 0AE14BA0     136              DC C        F0F4F3F0         0430
000144 0AE14BA4     137              DC C        F0F5F3F1         0531
000148 0AE14BA8     138              DC C        F0F6F3F0         0630
00014C 0AE14BAC     139              DC C        F0F7F3F1         0731
000150 0AE14BB0     140              DC C        F0F8F3F1         0831
000154 0AE14BB4     141              DC C        F0F9F3F0         0930
000158 0AE14BB8     142              DC C        F1F0F3F1         1031
00015C 0AE14BBC     143              DC C        F1F1F3F0         1130
000160 0AE14BC0     144              DC C        F1F2F3F1         1231
                    145    CHECKED_FOR_EOM_SW
000164 0AE14BC4                      DC C        E8               Y
                    146    PACKED_YEAR
000165 0AE14BC5                      DS PL3      00019F           ...
                    147    PACKED_YEAR2
000168 0AE14BC8                      DS PL2      0000             ..
00016A 0AE14BCA     148    XYEARS   DC P        0C               .
                                    DS CL1
                    149    CALC_EOM_RETURN@
00016C 0AE14BCC                      DS F        8AE14A98         ..6q
                         LTORG
000170 0AE14BD0     151                =C        F0F2             02
000172 0AE14BD2     152                =C        F2F8             28
000174 0AE14BD4     153                =C        F2F9             29
000176 0AE14BD6     154                =P        0C               .
000177 0AE14BD7     155                =P        4C               <

 STORAGE FOR DSECT #EOM_SW (R4)
Offset Address      Line  Label    Type        ----- HEX ------ - CHAR -
------ -------- --------- -------- --------    ---------------- --------
000000 0000F045     157    EOM_SW   DS C        D5               N

 STORAGE FOR DSECT #ANNIV_IND (R5)
Offset Address      Line  Label    Type        ----- HEX ------ - CHAR -
------ -------- --------- -------- --------    ---------------- --------
                    159    ANNIV_IND
000000 0000F046                      DS C        D5               N

 STORAGE FOR DSECT #RUN_DATE (R6)
Offset Address      Line  Label    Type        ----- HEX ------ - CHAR -
------ -------- --------- -------- --------    ---------------- --------
000000 0000F070     161    RUN_DATE DS 0CL6
000000 0000F070     162    RUN_YEAR DS CL2      F1F9             19
                    163    RUN_MONTH
000002 0000F072                      DS CL2      F9F8             98
000004 0000F074     164    RUN_DAY  DS CL2      F0F4             04
```

4. Review other supporting information.

   Whenever Assembler source listings are available in the source listing dataset, XLS provides the source listings for all active Assembler programs with the current instruction indicated. This information is shown in the Program Listing section. Figure 5-4 on page 5-5 shows the program listing for CWAADATE.

Figure 5-4. Program Listing Section

```
    ************************************************************************
    *              Program Listing Section - CWAADATE                      *
    ************************************************************************


   Stmt   Loc  Object Code       Source Statement
     42                        * Calculate years-of-service
     43                        * by subtracting employee year
     44                        * of hire from the current
     45                        * year.
     46                        ************************************************
     47                        CALC_ANNIVERSARY EQU *
     48                        * Uncomment following line before rolling
     49                        * this program into production!
     50                        *        ZAP   XYEARS,=P'3'
     51 00003C F221 B165 6000           PACK  PACKED_YEAR,RUN_YEAR
CURR => 000042 FD20 B165 B16A           DP    PACKED_YEAR,XYEARS
     53 000048 F900 B167 B176           CP    PACKED_YEAR+2(1),=P'0'
     54 00004E 4770 B064                BNE   EXIT
     55 000052 D501 7002 6002           CLC   HIRE_MONTH,RUN_MONTH
     56 000058 4770 B064                BNE   EXIT
     57 00005C 92E8 5000                MVI   ANNIV_IND,C'Y'
     58 000060 47F0 B064                B     EXIT
     60                        *********c**************************************
     61                        * Exit ***************************************
     62                        ************************************************
```

5. Review record information.

   The File section shows information about all files that are open at the time of the error. This information includes identification of the current record and previous records, when available. DCB and other control block information is also given. Abend-AID XLS displays records using decimal locations, making fields easier to find when using record layouts. File information is provided for all file types, including VSAM and IAM.

   You can directly access these files through File-AID from the online File Summary Selection List, which lists multiple open files. Inside a file you have full File-AID edit and browse capabilities.

Report Command

The View command in Assembler XLS enables you to see source data beyond column 72 of your terminal. The View setting continues throughout a session until changed. New sessions begin with the default View setting.

View Left (VL)

Default setting. Standard view from the line number, right 72 columns.

View Right (VR)

Compresses View Left data to the left and presents the complete label or any other data that was off screen in View Left.

Resolving an S0CB with Basic Language Support

When you access a report section through the Compuware/VF, your site's System Display and Search Facility (SDSF), or printed output — without the applicable Compuware language processor in use — the report does not provide source-level support. This comprises Abend-AID XLS basic language support.

Assembler basic support provides:

- The program name and the displacement of the instruction in error. Disassembly of the next sequential instruction and the instructions surrounding it.
- A call trace showing program names and displacements.
- Program storage for Assembler programs on the calling chain:
  - Formatted save area used to call each program
  - Storage of the actual CSECT
  - Formatted save area owned by each program.

The following procedure shows how to read a basic language report on the same error and program that was used in "Resolving an S0CB with Extended Language Support" on page 5-1.

1. Review the Error Analysis section to identify the error.

   When using Abend-AID XLS, first review the Error Analysis section, shown in Figure 5-5, for the analysis of the problem. Usually, all the information you need is in this section. The Error Analysis section also gives a definition of an S0CB abend. This example is a decimal divide exception, which is caused when the divisor is zero.

Figure 5-5. Error Analysis Section

```
   A decimal divide exception, S0CB, occurred during execution of program
   CWAADATE.

   ******************************************
   *          Analysis of Error             *
   ******************************************

   A decimal divide exception is caused when the divisor is zero.

                            Specific Information
                           Load
Absolute Program  DISPL/   Module   DISPL/   DISPL/        Machine
Address  Name     Prog     Name     EPA      Load      Len Instruction

0AE14AA2 CWAADATE 00000042 CWAADATE 00000042 00000042   6   FD20 B165 B16A
                                                         DP - divide decimal

                                                       A-operand - dividend

0AE14BC5 CWAADATE 00000165 CWAADATE 00000165 00000165   3   00019F

                                                       B-operand - divisor

0AE14BCA CWAADATE 0000016A CWAADATE 0000016A 0000016A   1   0C
                                              The above field is in error
```

2. Find the name of the program in error.

   Review the Error Location section in your Abend-AID XLS report. The Error Location section, shown in Figure 5-6 on page 5-7, identifies the CSECT name and the offset of the next sequential instruction to be executed.

   The Error Location section also provides the following information:

   - Program's compile date and length
   - Program's link date and load module length
   - Load module name and the load library name
   - Location of the last I/O operation or subroutine call, if applicable.
   - Disassembly of the next sequential instruction and the three instructions before and after it.

Figure 5-6. Error Location Section

```
*****************************************
*             Error Location            *
*****************************************

The next sequential instruction to be executed in program CWAADATE
was at displacement 00000048.

        00000038  92E8B164      MVI   356(R11),X'E8'
        0000003C  F221B1656000  PACK  357(3,R11),0(2,R6)
        00000042  FD20B165B16A  DP    357(3,R11),362(1,R11)
 NSI==> 00000048  F900B167B176  CP    359(1,R11),374(1,R11)
        0000004E  4770B064      BNE   100(,R11)
        00000052  D50170026002  CLC   2(2,R7),2(R6)
        00000058  4770B064      BNE   100(,R11)

The program was compiled on 08 FEB 2002 and is 00000178 bytes long.

It is part of load module CWAADATE.

The module was loaded from STEPLIB library CW.AA.TEMPLOA7.

The module was link edited on 08 FEB 2002 and is 00000178 bytes long.
```

3. Obtain a copy of the correct assembly listing.

   Make sure that your assembly listing matches exactly with the date provided by Abend-AID XLS. Look at the compile date provided in the Error Location section in Figure 5-6. Compare it to the date in the assembly listing shown in Figure 5-7. If you had multiple compiles on that date, then look at the compiled program length that is also provided in the Error Location section.

Figure 5-7. Assembly Listing

```
       Calculate years-of-service                                                                  Page    4
 Active Usings: PARMLIST,R3  #EOM_SW,R4  #ANNIV_IND,R5  #RUN_DATE,R6  #HIRE_DATE,R7  CWAADATE,R11
 Loc  Object Code    Addr1 Addr2  Stmt   Source Statement                          HLASM R2.0  2000/04/07 09.52
                                   25 *****************************************************************
                                   26 * Start of program logic ****************************************
                                   27 *****************************************************************
                                   30 *****************************************************************
                                   31 * Reset the employee anniversary indicator and check if we
                                   32 * have already tested for end-of-month.
                                   33 *****************************************************************
000028 92D5 5000      00000        34          MVI   ANNIV_IND,C'N'
00002C 95D5 B164      00164        35          CLI   CHECKED_FOR_EOM_SW,C'N'
000030 4770 B03C            0003C  36          BNE   CALC_ANNIVERSARY
000034 4DA0 B06E            0006E  37          BAS   10,CALC_EOM
000038 92E8 B164      00164        38          MVI   CHECKED_FOR_EOM_SW,C'Y'
                                   41 *****************************************************************
                                   42 * Calculate years-of-service by subtracting employee hire date
                                   43 * from current year.  If the years of service are divisible by
                                   44 * X, set the anniversary indicator on.  These employees will
                                   45 * be flagged on the employee compensation report.
                                   46 *****************************************************************
                            0003C  47 CALC_ANNIVERSARY EQU *
                                   48 * Uncomment following line before rolling
                                   49 * this program into production!
                                   50 *        ZAP   XYEARS,=P'3'            3-year anniversary
00003C F221 B165 6000 00165 00000  51          PACK  PACKED_YEAR,RUN_YEAR
000042 FD20 B165 B16A 00165 0016A  52          DP    PACKED_YEAR,XYEARS
000048 F900 B167 B176 00167 00176  53          CP    PACKED_YEAR+2(1),=P'0'  Remainder zero?
00004E 4770 B064            00064  54          BNE   EXIT                     No, not X-year anniversary
000052 D501 7002 6002 00002 00002  55          CLC   HIRE_MONTH,RUN_MONTH     Yes, same as this month?
000058 4770 B064            00064  56          BNE   EXIT                      No, not an anniversary
00005C 92E8 5000      00000        57          MVI   ANNIV_IND,C'Y'            Yes, set flag in caller
000060 47F0 B064            00064  58          B     EXIT
```

4. Look at the source code of the instruction in error.

   The Error Location section, shown in Figure 5-6, provides the hexadecimal displacement of the next sequential instruction. Look at the assembly listing in Figure 5-7 and find the next sequential instruction using this displacement. Typically, the instruction preceding the next sequential instruction is the instruction in error.

5. Determine the operand in error.

   The instruction in error contains two fields, as shown in Figure 5-7 on page 5-7. Refer again to the Error Analysis section, shown in Figure 5-5 on page 5-6. Abend-AID XLS identifies the cause of the error and the field causing the decimal divide exception. The Specific Information area of the Error Analysis section indicates that the B-operand is the field in error. This variable contains X'0C'. The divisor in the instruction in error is XYEARS.

Figure 5-8. Supporting Environmental Data Section

```
   ******************************************
   *      Supporting Environmental Data     *
   ******************************************

   PARM Data - 00003
   Length of data -    5

 Abend PSW - 078D1000 8AE14AA8   A(CWAADATE) + 00000048
 Prog PSW  - 078D1000 8AE14AA8   A(CWAADATE) + 00000048
 Load Module - CWAADATE                  Entry Point Address - 0AE14A60
 ILC - 06,   INTC - 0B                   Load  Point Address - 0AE14A60

  Registers at time of error  (Descriptions based on 31 bit addresses)

REG HEX       Decimal         Description
R0  00030320          197,408  A(0002C000) + 00004320, PVT SP=001 ALLOC
R1  0000EAD0           60,112  A(CWAACOB ) + 00008A30
R2  00021719          136,985  A(00021000) + 00000719, PVT SP=001 ALLOC
R3  0000EAD0           60,112  A(CWAACOB ) + 00008A30
R4  0000F045           61,509  A(CWAACOB ) + 00008FA5
R5  0000F046           61,510  A(CWAACOB ) + 00008FA6
R6  0000F070           61,552  A(CWAACOB ) + 00008FD0
R7  00010155           65,877  A(CWAACOB ) + 0000A0B5
R8  00000000                0
R9  0AE14BC4     182,537,156  A(CWAADATE) + 00000164
R10 8AE14A98  -1,964,946,792  A(CWAADATE) + 00000038
R11 0AE14A60     182,536,800  A(CWAADATE) + 00000000
R12 0001F6A0         128,672  A(0001D000) + 000026A0, PVT SP=001 ALLOC
R13 0AE14B4C     182,537,036  A(CWAADATE) + 000000EC
R14 800244A8  -2,147,335,000  A(IGZCLNK ) + 00000410
R15 8AE14A60  -1,964,946,848  A(CWAADATE) + 00000000

                  Floating point registers at time of error

        R0                R2                R4                R6
 4930D7E5A8000000  4E000000021F6F28  4E0000000000250C7  0000000000000000
```

6. Find the contents of other variables in your program.

   Use Abend-AID XLS to investigate the contents of other variables in your program. Find the contents of PACKED_YEAR. In Figure 5-7 on page 5-7, the assembly listing indicates that this field is based off of register 11 at a displacement of X'165' bytes and has a length of three bytes. In Figure 5-8, the address in register 11 is shown in the Supporting Environmental Data section of the Abend-AID XLS report. Adding the field's displacement of X'165' bytes to this address X'0AE14A60' yields X'0AE14BC5' which is the actual address of the field. In Figure 5-9 on page 5-9, the contents of field PACKED_YEAR are found in the three bytes of data starting at this address.

Figure 5-9. Program Storage for CSECT CWAADATE

```
                    ************************************
                    * Program storage for CSECT CWAADATE *
                    * located in load module CWAADATE    *
                    * Length is 00000178                 *
                    * Language is HLASM                  *
                    ************************************

*********************************************************************
*               Save area used to call CSECT CWAADATE               *
*********************************************************************
* Located at:  00030018  A(0002C000) + 00004018, PVT SP=001 ALLOC  *
* Descriptions based on 31 bit addresses                            *
*********************************************************************
    HEX       Decimal         Description
WD1 00102401       1,057,793  UNALLOCATED - LSQA
BCK 0000E858          59,480  A(CWAAC0B ) + 000087B8
FWD 00030320         197,408  A(0002C000) + 00004320, PVT SP=001 ALLOC
R14 800244A8  -2,147,335,000  A(IGZCLNK ) + 00000410
R15 8AE14A60  -1,964,946,848  A(CWAADATE) + 00000000
R0  00030320         197,408  A(0002C000) + 00004320, PVT SP=001 ALLOC
R1  0000EAD0          60,112  A(CWAAC0B ) + 00008A30
R2  00021719         136,985  A(00021000) + 00000719, PVT SP=001 ALLOC
R3  0000EAD0          60,112  A(CWAAC0B ) + 00008A30
R4  0000654B          25,931  A(CWAAC0B ) + 000004AB
R5  00000004               4
R6  000507B8         329,656  A(0002C000) + 000247B8, PVT SP=001 ALLOC
R7  0000E9B0          59,824  A(CWAAC0B ) + 00008910
R8  800244AA  -2,147,334,998  A(IGZCLNK ) + 00000412
R9  0000E858          59,480  A(CWAAC0B ) + 000087B8
R10 00050038         327,736  A(0002C000) + 00024038, PVT SP=001 ALLOC
R11 80024098  -2,147,336,040  A(IGZCLNK ) + 00000000
R12 0001F6A0         128,672  A(0001D000) + 000026A0, PVT SP=001 ALLOC

*********************************************************************
*                    Storage of CSECT CWAADATE                      *
*********************************************************************
 DSPL Address   -------------- Data ---------------
00000 0AE14A60  90ECD00C 18BF41B0 B00050D0 B0F041D0  ..}.......&}.0.}
00010 0AE14A70  B0EC1831 58403000 58503004 58603008  ..... ...&...-..
00020 0AE14A80  5870300C 41707000 92D55000 95D5B164  ........kN&.nN..
00030 0AE14A90  4770B03C 4DA0B06E 92E8B164 F221B165  ....(..>kY..2...
00040 0AE14AA0  6000FD20 B165B16A F900B167 B1764770  -.......-9.......
00050 0AE14AB0  B064D501 70026002 4770B064 92E85000  ..N...-.....kY&.
00060 0AE14AC0  47F0B064 58D0B0F0 98ECD00C 07FE50A0  .0...}.Oq.}...&.
00070 0AE14AD0  B16CD501 6002B170 4780B0AC 4180000C  .%N.-...........
00080 0AE14AE0  4190B134 D5019000 60024780 B09A4190  ....N...-.......
00090 0AE14AF0  90044680 B08447F0 B0ACD501 90026004  .....d.0..N...-.
000A0 0AE14B00  4770B0A8 92E84000 47F0B0E6 F221B165  ...ykY ..0.W2...
000B0 0AE14B10  6000FD20 B165B177 F900B167 B1764780  -.......9.......
000C0 0AE14B20  B0D4D501 6004B172 4770B0E6 92E84000  .MN.-......WkY .
000D0 0AE14B30  47F0B0E6 D5016004 B1744770 B0E692E8  .0.WN.-......WkY
000E0 0AE14B40  400047F0 B0E658A0 B16C07FA 00000000   ..0.W...%......
000F0 0AE14B50  00030018 00000000 00000000 00000000  ................
00100 0AE14B60  00000000 00000000 00000000 00000000  ................
            LINES 0AE14B70-0AE14B80 SAME AS ABOVE
00130 0AE14B90  00000000 F0F1F3F1 F0F2F2F8 F0F3F3F1  ....013102280331
00140 0AE14BA0  F0F4F3F0 F0F5F3F1 F0F6F3F0 F0F7F3F1  0430053106300731
00150 0AE14BB0  F0F8F3F1 F0F9F3F0 F1F0F3F1 F1F1F3F0  0831093010311130
00160 0AE14BC0  F1F2F3F1 E800019F 00000C00 8AE14A98  1231Y.........Ûq
00170 0AE14BD0  F0F2F2F8 F2F90C4C                    022829.<
```

7. Review record information.

   The File section shows information about all files that are open at the time of the error. This information includes identification of the current record and previous records, when available. DCB and other control block information is also given. Abend-AID XLS displays records using decimal locations, making fields easier to find when using record layouts. File information is provided for all file types, including VSAM and IAM.

   You can directly access these files through File-AID from the online File Summary Selection List, which lists multiple open files. Inside a file you have full File-AID edit and browse capabilities.

Chapter 6.
Diagnosing VS FORTRAN Errors

This chapter summarizes the use of Abend-AID XLS to diagnose VS FORTRAN errors. For VS FORTRAN, Abend-AID XLS provides only basic language support.

VS FORTRAN Diagnostics

Abend-AID XLS helps FORTRAN programmers diagnose application program errors by reporting information that the VS FORTRAN normal error recovery process obscures. VS FORTRAN run time routines intercept application program abends when its SPIE and STAE run-time options are in effect. VS FORTRAN then terminates the application with a U0240 abend, hiding the original error. Although basic information about the application's problem may be provided by VS FORTRAN messages, usually this information is insufficient in helping to solve errors. However, upon installation of Abend-AID XLS VS FORTRAN support, crucial diagnostic information pertaining to the original error is presented by Abend-AID XLS following the U0240 abend.

Abend-AID XLS provides extensive diagnostic capabilities for VS FORTRAN. Figure 6-1 shows a sample Error Analysis section for a VS FORTRAN program.

Figure 6-1. VS FORTRAN Error Analysis Section

```
A storage reference exception, SOC4, occurred during execution of program
FORTTST.

*****************************************
*          Analysis of Error            *
*****************************************

The system completion code of OC4 is issued when the program attempts to
use storage that is not accessible.


*** List of Probable OC4 Causes ***

1.  Missing or incorrect DD statement -
    Check system messages from abending step.

2.  Tried to read or write an unopened file.

3.  Tried to reference record area before file opened.

4.  Indexing or subscripting outside defined limits.
```

Abend-AID XLS automatically handles all U0240 abends by identifying the actual problem that caused the abend. It displays the complete CSECT and fully traces any called subroutines. Abend-AID XLS's full-function abend support does not inhibit the normal VS FORTRAN diagnostic capabilities.

VS FORTRAN Extended Error Handling Facility

Abend-AID XLS VS FORTRAN support does not interfere with the VS FORTRAN normal error recovery and reporting process. This means that VS FORTRAN may allow a number of errors of a particular type to occur in an application before taking corrective action or terminating the application with a U0240 abend. Therefore, if an application program

encounters errors but completes before the error count is exceeded, no U0240 abend occurs. Consequently, no Abend-AID XLS report is produced.

FORTRAN programmers may find that they need lower error counts in order to obtain Abend-AID XLS reports for certain program errors. The site can modify option tables to change default error counts for all VS FORTRAN application programs running on a system. In addition, an individual program may change error counts in effect for the duration of a single application run by calling ERRSET. These VS FORTRAN facilities are described in the IBM manuals *VS FORTRAN Language and Library Reference* and *VS FORTRAN Programming Guide — CMS, MVS*.

Chapter 7.
Understanding Database Diagnostics

This chapter summarizes the features of Abend-AID XLS's database diagnostics:

- Abend-AID for DB2
- Abend-AID for IMS
- Abend-AID for IDMS.

Note: Database files identified in the File section of a report can be accessed through File-AID. Refer to "File Section" on page 12-19 for information.

Abend-AID for DB2

Abend-AID for DB2 provides support for IBM's relational database manager, DB2. It presents the information needed to solve not only everyday but also unique, complex DB2 abends in a few concise report pages.

The Abend-AID for DB2 report provides a diagnostic report containing information about the statement in error and the specific SQL command that failed. It identifies the cause of the error and suggests a course of action.

A comprehensive DB2 Environment section under TSO, CALL, or RRSAF (Recoverable Resource Manager Services Attachment Facility) attach mode processing contains a summary of information to assist you in resolving the problem. This section includes:

- SQL code interpretation
- SQL statement in error
- Interpreted contents of the SQLCA
- Contents of the SQLDA, for dynamic SQL
- Display of the columns referenced
- Names and contents of the host variables
- Plan name, bind time, and bind date
- Package bind information and package precompile information
- DB2 release, subsystem, and authorization ID
- Related information.

The Abend-AID for DB2 report also provides abend return codes with corresponding explanations. This information is especially useful for resolving S04E reason codes.

Figure 7-1 on page 7-2 illustrates an Abend-AID XLS for DB2 Error Analysis section for an S04E abend.

Figure 7-1. Abend-AID for DB2 Error Analysis Section – S04E Abend

```
An S04E abend occurred during execution of program INVDB01.

****************************************
*          Analysis of Error           *
****************************************

DB2 has returned a 00C90101 reason code after a S04E    abend

The data manager detected an internal error within DB2.  This error
may indicate inconsistent data or an error in internal DB2 control
structure or code.
Probable causes:
     1.  Improper migration or fallback procedures.
     2.  DB2 directory and DB2 catalog restored to different
         points in time.
     3.  Table space restored improperly.
     4.  An internal DB2 failure.
     Notify your systems programmer.

The SQL statement returning this SQLCODE at preprocessor sequence
number    93 in program 'TST04E  ' is:

EXEC SQL FETCH TEMPL_CURSOR
    INTO : DCLEMP3

The SQL statement that defined the cursor TEMPL_CURSOR
at preprocessor sequence number    69 is:

EXEC SQL OPEN TEMPL_CURSOR

The SQL statement that defined the cursor TEMPL_CURSOR
at preprocessor sequence number    62 is:

EXEC SQL DECLARE TEMPL_CURSOR
    CURSOR FOR
    SELECT * FROM CWXAAD2 . SAMPEMP
    WHERE LASTNAME = 'HAAS'

See the DB2 environment data below for further details.
```

Using Report Types

Four different situations will produce an Abend-AID XLS report with DB2 support. Each of these situations produces a report with slightly different types of information. Different parts of the Abend-AID XLS report are significant. The following describes the four situations and corresponding importance of the Error Analysis, Error Location, and DB2 sections of the Abend-AID XLS report. Other sections do not differ among the four report types and should be viewed as needed.

- Non-DB2 system error occurs in a DB2 program:
  - Error Analysis section is important and should be viewed first.
  - Error Location section contains useful information about the location of the error.
  - DB2 section should be checked for supporting information (for example, the last SQL code returned).
- DB2 system abend occurs (S04E, S04F):
  - Error Analysis section is important and should be viewed first. Location of the error is found here.
  - Error Location section is not pertinent because the abend does not occur in the application program.
  - DB2 section should always be viewed.
- Negative SQL code – user abend is forced:
  - Error Analysis section is significant only if customized diagnostic messages are provided for the user abend.

  - Error Location section is not pertinent because the abend does not occur in the application program.
  - DB2 section is most valuable. Location and description of the error is found here. View this section first.

- Negative SQL code – SNAP-AID called by the application program:
  - Error Analysis section is not present because no abend occurs.
  - Error Location section points to the application program, but the location is the call to SNAP-AID, not where the DB2 error occurred.
  - DB2 section is most valuable. Location and description of the error is found here. View this section first.

If your shop standards don't require the forcing of user abends for negative SQL codes, you should consider using SNAP-AID. The program can then be terminated or continue to execute based on the severity of the error.

Using SNAP-AID to Report Negative SQL Codes

Abend-AID XLS is automatically invoked when an abend occurs. However, when a negative SQL code is encountered, an abend does not occur and program processing continues. Abend-AID for DB2 provides support for more than 500 SQL codes. To obtain complete DB2 information at various points in the program without terminating the program, use SNAP-AID. A simple call to the SNAPAID module in the application program is all that is needed to invoke this support. The following three figures show examples of a SNAP-AID call in an application program.

In Figure 7-2 and Figure 7-3, the SQLCODE is a field in the SQL Communications Area (SQLCA) that contains the last SQL return code. The RETURN-CODE is the step's completion code that appears in the job log. It can be used to indicate if further steps of the job should be run.

Figure 7-2. SNAP-AID Example for a COBOL Program

```
PROCEDURE DIVISION.
     EXEC SQL WHENEVER SQLERROR GO TO 9999-SQLERROR END-EXEC.

9999-SQLERROR.
     CALL 'SNAPAID'.
     MULTIPLY SQLCODE BY -1 GIVING RETURN-CODE
     GOBACK.
```

Figure 7-3. SNAP-AID Example for a PL/I Program

```
ERROR_CHECK:   PROC;
     EXEC SQL WHENEVER SQLERROR GO TO 9999_SQLERROR;
END ERROR_CHECK;

9999-SQLERROR:    PROC;
     CALL 'SNAPAID'
     RETURN_CODE = SQLCODE * -1;
END 9999_SQLERROR;
```

The method shown in Figure 7-4 preserves structured programming and brings the call to SNAPAID closer to the statement that produced the error.

Figure 7-4. SNAP-AID Example for a COBOL Program

```
8888-GET-RECORD.
      EXEC SQL
          SELECT ....
            INTO ....
            FROM ....
            WHERE....
      END EXEC

      IF SQLCODE EQUAL ZERO
           NEXT SENTENCE
      ELSE IF SQLCODE EQUAL +100
           MOVE "Y" TO NO-ROWS-SWITCH
      ELSE
           CALL 'SNAPAID'
           PERFORM 9999-SQLERROR.
```

Many shops prefer to issue user abends when negative SQL codes indicate severe errors. The following are examples of coding user abend codes.

In Figure 7-5, the SQLCODE is a field in the SQLCA that contains the last SQL return code. 'ILBOABN0' is the COBOL-supplied routine to force an abend. A user-written routine that produces a dump can be used instead of 'ILBOABN0'.

Figure 7-5. User Abend Example for a COBOL Program

```
PROCEDURE DIVISION.
    EXEC SQL WHENEVER SQLERROR GO TO 9999-SQLERROR END-EXEC.

9999-SQLERROR.
    MULTIPLY SQLCODE BY -1 GIVING RETURN-CODE
    CALL 'ILBOABN0' USING RETURN-CODE.
```

In Figure 7-6, the SQLCODE is a field in the SQLCA that contains the last SQL return code. PLIRETC is the PL/I-supplied routine to set a return code.

Figure 7-6. User Abend Example for a PL/I Program

```
ERROR_CHECK:    PROC;
     EXEC DQL WHENEVER SQLERROR GO TO 9999_SQLERROR;
END ERROR_CHECK;

9999_SQLERROR:   PROC;
     RETURN_CODE = SQLCODE * -1;
     CALL PLIRETC (RETURN_CODE);
END 9999_SQLERROR;
```

Refer to **Chapter 8, "Using SNAP-AID"** for additional information on SNAP-AID.

Note: Abend-AID for DB2 uses the call and RRSAF attach interfaces to DB2. The load modules for these attach facilities must be available to Abend-AID for DB2 during the link-edit part of the installation and at execution.

With TSO, Call, or RRSAF Attach Mode

Whenever an Abend-AID XLS report is produced for a DB2 program, diagnostic information about the last SQL completion code is provided in the DB2 Environment Data section. The DB2 support does not require any language processing to pinpoint the exact SQL statement and the host variable names.

The DB2 Environment Data section combines IBM dump, SQLCA, and DB2 catalog information to provide essential data needed to resolve infrequent but complex DB2 errors.

In the TSO, call, or RRSAF attach mode environment, the following information is also provided:

- Full text and statement number of SQL statement in error as well as any related SQL statements
- The source data name of all host variables referenced
- When appropriate and required, the name, type, and table name of all columns referenced
- Bind options, date, and time
- Precompile options
- DBRM library name
- Names and types of all DB2 objects on which the Application Plan is dependent
- DB2 release, authorization ID, subsystem ID, plan name, and DBRM name
- Detailed Plan Package data, including bind, precompile, and dependency information.

Figure 7-7 on page 7-5 through Figure 7-10 on page 7-7 show examples of the Abend-AID for DB2 Environment section under TSO, call, or RRSAF attach mode processing.

Figure 7-7. TSO/Call/RRSAF Attach Mode – SQL Code and Statement in Error

```
                      DB2 Environment Data
                      --------------------
                       AADB2 Release 5.0.6

                       SQL Return Code
                       ---------------
SQLCODE     -302

The value of input variable is too large for the target column.

Following is a list of those host variables that are too large
for their matching columns.

Host Variable Name/
Column Name/
Table Name                          Type
-----------                         ----
:WS-FIELDS.WS-ACCOUNT               DECIMAL(015,006)

DESC_ACCOUNT                        DECIMAL(005,000)
DESC

The SQL statement returning this SQLCODE at preprocessor sequence
number    88 in program 'DTC2302S' is:

EXEC SQL INSERT
    INTO CWXAAD2 . DESC
    ( DESC_ACCOUNT ,
    DESC_PRODUCT ,
    DESC_ACTION )
    VALUES
    ( : WS-FIELDS.WS-ACCOUNT ,
    : WS-FIELDS.WS-PRODUCT ,
    : WS-FIELDS.WS-ACTION )
```

Figure 7-8. TSO/Call/RRSAF Attach Mode – General Information and Host Variables Referenced

```
                        DB2 General Information
                        -----------------------

   DB2 Release:      4.1.0                Subsystem ID: D41A
   Authorization ID: EFHJFGO              Attach Mode:  TSO

                   Application Plan Name: DTC2302

   Bind Date:        18 OCT 2000          Bind Time:    14:27:40.89
   Isolation:        CURSOR STABILITY     Acquire:      USE
   Release:          COMMIT               Validate:     BIND

                   Database Request Module (DBRM): DTC2302S

   Precompile Date: 18 OCT 2000           Precompile Time: 18:27:11.90
   Host Language:   COBOL
   SQL Escape:      ' (APOSTSQL)          SQL Decimal:     . (PERIOD)

   Library Name: D2.SAMPLES.V2R3M0.DBRM

                         Host Variables Referenced
                         -------------------------
   Field                              Type                      Values
   -----                              ----                      ------
   :WS-FIELDS.WS-ACCOUNT              DECIMAL(015,006)    123456789.123456

   :WS-FIELDS.WS-PRODUCT              CHAR(030)               Below

0004BF67   CHAR  12345678901234567890123456789O
           ZONE  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
          DIGIT  12345678901234567890123456789O
                 1...5...10....*...20....*...30

   :WS-FIELDS.WS-ACTION               CHAR(024)               Below

0004BF87   CHAR
           ZONE  444444444444444444444444
          DIGIT  OOOOOOOOOOOOOOOOOOOOOOOO
                 1...5...10....*...20....
```

Figure 7-9. TSO/Call/RRSAF Attach Mode – Columns Referenced

```
                          Columns referenced
                          ------------------
     Column Name/                        Type
     Table Name
     -----------                         ----
     DESC_ACCOUNT                        DECIMAL(005,000)
     DESC

     DESC_PRODUCT                        VARCHAR(00024)
     DESC

     DESC_ACTION                         VARCHAR(00024)
     DESC
```

Figure 7-10. TSO/Call/RRSAF Attach Mode – SQLCA and Plan Dependencies

```
                        SQL Communications Area (SQLCA)
                        -------------------------------

SQLCAID     SQLCA
SQLCABC       136
SQLCODE      -302
SQLERRML        4
SQLERRMC
001.
SQLERRP     DSNXRBND
SQLERRD(1)              -740      X'FFFFFD1C'
SQLERRD(2)                 0      X'00000000'
SQLERRD(3)                 0      X'00000000'
SQLERRD(4)                -1      X'FFFFFFFF'
SQLERRD(5)                 0      X'00000000'
SQLERRD(6)                 0      X'00000000'
SQLWARN0
SQLWARN1
SQLWARN2
SQLWARN3
SQLWARN4
SQLWARN5
SQLWARN6
SQLWARN7
SQLSTATE       22001
                      ------------------

Object Qualifier               Object Name       Object Type
----------------               -----------       -----------
  CWXAAD2                   DESC                 TABLE
   AAD41ADB                  AAD41ATS             TABLE SPACE
```

With IMS or DL/I Attach Mode

In IMS or DL/I attach mode, the DB2 catalog cannot be accessed because of environment limitations. Therefore, the reports generated in these environments are slightly different from those generated for TSO, call, or RRSAF attach mode.

The DB2 Environment Data section formats IBM dump and SQLCA information to provide essential data needed to resolve complex DB2 errors.

In the IMS or DL/I attach mode environment, the following information is provided:

- Statement number and verb of the SQL statement in error
- DB2 release, authorization ID, subsystem ID, plan name, and DBRM name
- COBOL base locator cell and offset of all host variables referenced in the SQL statement in error. When XLS is used, the host variable names are provided instead of base locator cell/offset.

Figure 7-11 on page 7-8 shows the SQL Return Code and DB2 General Information areas of the Abend-AID for DB2 Environment Data section. These areas identify the SQL verb in error for the SQL statement returning the SQL code.

Figure 7-11. IMS or DL/I Attach Mode – SQL Code and Verb in Error

```
                        DB2 Environment Data
                        --------------------

                          SQL Return Code
                          ---------------

    SQLCODE     -811

    One of the following has occurred:

    1.      The execution of a 'SELECT...INTO' statement has returned a
            result table containing more than one row.

    2.      A subquery contained in a basic predicate has produced more
            than one value.

    Corrective Action:
       Examine the syntax of the statement to ensure that it contains the
       proper condition specifications.  If it does, there may be a problem
       with the data that is causing more than one row or value to be
       returned when you do not expect it.

    The SQL statement returning this SQLCODE at preprocessor sequence
    number  2093 in program 'LUN8000U' is:

    EXEC SQL SELECT

                        DB2 General Information
                        -----------------------
    DB2 Release:      3.1.0                Subsystem ID: DSP2
    Authorization ID: P011520              Attach Mode:  IMS

                    Application Plan Name: LUN8000U

                    Database Request Module (DBRM): LUN8000U
    Precompile Date: 15 JAN 2000             Precompile Time: 13:11:36.35
```

Figure 7-12 shows the Host Variables Referenced area of the Abend-AID XLS report. Because the variable names cannot be accessed from the DB2 catalog, this area includes the COBOL base locator cell and offset of the host variables; or for other languages, the relative addresses.

Figure 7-12. IMS or DL/I Attach Mode – Host Variables Referenced

```
                        Host Variables Referenced
                        -------------------------
   Field                              Type                      Values
   -----                              ----                    ------------
   OFFSET=0D2C IN BLW     9 (X'9')    DECIMAL(009,000)           222222222

   OFFSET=0D31 IN BLW     9 (X'9')    DECIMAL(005,000)                3401

   OFFSET=0D34 IN BLW     9 (X'9')    CHAR(010)                Below

000FDD5C   CHAR  12 20   86
           ZONE  FF0FF000FF
          DIGIT  1202000086
                 1...5...10
```

If COBOL XLS is used, Abend-AID for DB2 can provide the data name for each host variable. Figure 7-13 shows the same host variables when COBOL XLS is used.

Figure 7-13. IMS or DL/I Attach Mode – Host Variables Referenced Using COBOL XLS

```
                        Host Variables Referenced
                        -------------------------

   Field                                Type                      Values
   -----                                ----                   ------------
   :WS-SSN                              DECIMAL(009,000)          222222222

   :WS-ACCOUNT                          DECIMAL(005,000)               3401

   :WS-DATE                             CHAR(010)              Below

000FDD5C   CHAR  12 20   86
           ZONE  FF0FF000FF
          DIGIT  1202000086
                 1...5...10
```

Using DB2 Plan Package Support

With DB2 Version 2 Release 3, IBM introduced DB2 Plan Packages, reducing the complexity of DB2 plan management and eliminating time-consuming rebinds of large plans. Abend-AID for DB2 Plan Package support provides the following additional information in the DB2 section of the Abend-AID XLS report:

- Package information
- Package bind information
- Package precompile information
- Package dependencies.

Figure 7-14 shows the DB2 General Information area of an Abend-AID for DB2 report with plan package support.

Figure 7-14. DB2 General Information Section

```
                        Package Bind Information
                        ------------------------
   Location:          DEFAULT          Collection-id:   AATCVCOL
   Name:              DTCV508S         Qualifier:       CWX0070
   Owner:             CWX0070          Creator:         CWX0070
   Bind Date:         03 MAR 2000      Bind Time:       10:06:05.01
   Create Date:       02 MAR 2000      Create Time:     15:42:41.87
   Isolation:         CURSOR STABILITY Validate:        BIND
   Release:           COMMIT           SQL Error:       NOPACKAGE
   Valid Package:     YES              Operative:       YES
   Current Data:      NO

                     Package Precompile Information
                     ------------------------------

   Precompile Date:    03 MAR 2000     Precompile Time: 09:59:01.14
   Precompile Version: AADB2TST
   Host Language:      COBOL           DEC31:           NO
   SQL Escape:         ' (APOSTSQL)    SQL Decimal:     . (PERIOD)
   Mixed Data:         NO              Katakana:        NO
   Precompile Release: 2.2.0

   Library Name:       CW.D2.SAMPLES.V3R1M0.DBRM
```

Using DB2 Stored Procedure Support

Abend-AID for DB2 provides diagnostic information for stored procedures in Language Environment (LE). Stored procedures (user-written applications) run in a special DB2 stored procedure address space. It is difficult to diagnose problems because only the SQLCODE is returned to the calling program. Abend-AID for DB2 provides diagnostic information for easy problem resolution.

Figure 7-15 shows Abend-AID for DB2 diagnostic information when the application is a stored procedure.

Figure 7-15. DB2 Diagnostic and General Information Section

```
                    DB2 Environment Data
                     --------------------
                      AADB2 Release 5.0.6


                      SQL Return Code
                      ---------------
SQLCODE     -407

An 'UPDATE' or 'INSERT' value or the value in a predicate is NULL, but
the object column cannot contain NULL values.

Corrective action:
  Examine the definition of the object table to determine which
  columns of the table have the NOT NULL attribute, and correct
  the SQL statement.

The SQL statement returning this SQLCODE at preprocessor sequence
number    79 in program 'AADPSP2 ' is:

EXEC SQL UPDATE
    TEMPL
    SET
    WORKDEPT = : DCLTEMPL.WRKDEPT : DCLTEMPL.WRKDEPTI WHERE WORKDEPT =
    '001'

                      DB2 General Information
                      -----------------------

DB2 Release:      4.1.0               Subsystem ID:  D410
Authorization ID: D410SPA             Attach Mode:   CALL


                          Plan Information
                          ----------------

                 Application is a Stored Procedure
```

Supported Reason Codes

Table 7-1 shows the number of reason codes, by DB2 subcomponent, that are supported by Abend-AID for DB2.

Table 7-1. Abend Reason Code Support Information

DB2 Subcomponent	Hex ID	Abend-AID for DB2 Support	# of Reason Codes
Call Attachment Facility	X'C1'	Yes	54
Buffer Manager	X'C2'	Yes	116
CICS Attachment Facility	X'C3'	Yes	14
TSO Attachment Facility	X'C5'	Yes	16
Message Generator	X'C6'		
Precompiler, DSNH	X'C8'	Yes	10
Data Manager	X'C9'	Yes	101
Recovery Log Manager	X'D1'	Yes	155
Distributed Data Facility	X'D3'		
IMS/VS Attachment Facility	X'D4'	Yes	29
DB2I Help Panel	X'D5'		
Data Space Manager	X'D7'	Yes	34
Recovery Manager 5+	X'D9'	Yes	67
Storage Manager	X'E2'	Yes	47
Service Controller	X'E3'	Yes	76
Utilities	X'E4'		

Table 7-1. Abend Reason Code Support Information

DB2 Subcomponent	Hex ID	Abend-AID for DB2 Support	# of Reason Codes
Agent Services Manager	X'E5'	Yes	84
Instrumentation Facility	X'E6'	Yes	72
Relational Data System	X'E7'	Yes	50
Initialization Procedures	X'E8'	Yes	61
System Parameter Manager	X'E9'	Yes	5
Service Facilities	X'F1'	Yes	2
Subsystem Support Subcomponent	X'F3'	Yes	202
Sample Applications	X'F8'		
General Command Processor	X'F9'	Yes	14
Total			1261

ROLLBACK Statement

Use the DB2 ROLLBACK statement to terminate the current unit of recovery, back out the database changes that were made within that unit of recovery, and start a new unit of recovery. Compuware considers ROLLBACK to be an important part of error recovery processing. Note the method that Abend-AID XLS uses to handle DB2 diagnosis during ROLLBACK.

When a ROLLBACK occurs, the control blocks that Abend-AID XLS relies on to diagnose DB2 problems are freed. If Abend-AID XLS is invoked after a ROLLBACK, Abend-AID XLS finds the control blocks for the new unit of recovery and reports that the DB2 connection was not utilized. For ROLLBACK recovery, Compuware recommends a call to the module SNAPAID before the ROLLBACK is issued. SNAP-AID allows Abend-AID XLS to properly diagnose the DB2 problem and return control back to the program. When the program regains control, you can issue the ROLLBACK, and the integrity of the database is preserved.

File-AID Access

From the DB2 section you can directly access a DB2 database through File-AID/DB2. Edit and Browse commands in the report take you into File-AID edit and browse modes within the database. A current version of File-AID/DB2 must be available on the operating system.

Abend-AID for IMS

Abend-AID for IMS provides information about the IMS DB and TM environments, the application program, and the abending program's PCBs and JCBs. In addition, the last six calls to the database are listed. Abend-AID for IMS diagnoses all IMS user abends and gives both the cause of the error and the solution. Additional information such as the contents of DFSPRPX0 and DFSECP is provided, if appropriate.

Figure 7-16 through Figure 7-19 on page 7-13 show Abend-AID for IMS examples.

Figure 7-16. Abend-AID for IMS Section – COBOL XLS

```
                            IMS-AID Diagnostics

 Application program name: PAYIMSLP           IMS Rel:510  I510
 Calling PARM FORMAT:COBOL                    Region type: DLI Batch
 PSB Name: PSBCBLBT

    The last IMS call was:
000160                    EM-PCB
000161                    EM1-ROOT
000162                    EMROOT-SSA.

    This statement is contained in paragraph "P200-UPDATE-DB" of
    program PAYIMSLP which is 00001180 bytes long.   This program is
    part of load MODULE PAYIMSLP which is 00002828 bytes long.
*************************************************************************
    To assist the programmer, the following key IMS control blocks
    are printed.

    1. Program Communications Block (PCB).  one for each data base
    being accessed.

    2. Control block and table module DFSPRPX0.

    Up to the last 6 calls against the data base as logged in each
    JCB will be printed.

    If multiple PCBs are used, the current PCB (if any), is labeled and
    printed first.
*************************************************************************
```

Figure 7-17. Abend-AID for IMS Section – COBOL Basic Support

```
                            IMS-AID Diagnostics

   Application Program Name: PCBTEST         IMS Rel: 510 IMS2
   Calling PARM FORMAT: COBOL                Region Type: DLI BATCH
   PSB Name: PSBCBLBT

           The last IMS call was made at displacement 00003C in
           program IBMBDLI1 which is 0000008C bytes long and is part
           of load module PCBTEST which is 00008538 bytes long.
   ************************************************************************
      To assist the programmer, the following key IMS control blocks
      are printed.

      1. Program Communication Block (PCB).  One for each data base
      being accessed.

      2. Control block and table module DFSPRPX0.

      Up to the last 6 calls against the data base as logged in each
      JCB will be printed.

      If multiple PCBs used, the current PCB (if any) is labeled and
      printed first.
   ************************************************************************
```

Figure 7-18. Abend-AID for IMS Section – Database Calls (Batch Statistics)

```
          DL/I Call Statistics (from PST accounting area)

   DL/I Database Calls
      GU      GN      GNP      GNU      GHN      GHNP     ISRT     DLET     REPL
   1234567 1234568 1234569 1234570 1234571 1234572 1234573 1234574 1234575

   Total DLI database calls 11111139

   DL/I Message Calls
      GU       GN     ISRT     PURGE     CMD      GCMD
   1234577 1234578 1234579 1234580 1234581 1234582

   DL/I System Service Calls
     APSB     DPSB     CMSG     ICMD     RCMD     CHKP     XRST
   1234586 1234587 1234588 1234589 1234590 1234591 1234592

     ROLB     ROLS     SETS     SETU     INIT     INQY     LOG
   1234593 1234594 1234595 1234596 1234597 1234598 1234599
```

Figure 7-19. Abend-AID for IMS PCB and Call Trace Summary

```
                              ***************
                              * Current PCB *
                              * Function=GU *
                              ***************

    PCB       Database  Segment Status Process  Segment   KEY   Number of
    Address   Name      Level   Code   Options  Name      Len   Segments

    00028124  EMPLOYDB  01      'GE'   AP       EMROOT      4      2


    PCB Name=EMPLPCB   PCB Relative Number=  4


    KEY feedback From last call
00028148   CHAR  1222
           ZONE  FFFF
          DIGIT  1222
                 1...
                        JCB Database Call Trace

     # Call Type         Status Code and Description

         GU-GHU            GE-Segment not found
    - 1 GU-GHU            GE-Segment not found
    - 2 GU-GHU              -Normal return from call
    - 3 GU-GHU              -Normal return from call
    - 4 GU-GHU              -Normal return from call
    - 5 GU-GHU              -Normal return from call

Display of SSA number     1

000A5588   CHAR  EMROOT  *--
           ZONE  CDDDDE44566
          DIGIT  54966300C00
                 1...5...10.

Display of SSA number     2

000A5620   CHAR  EMSALARY*--(EMSALKEY =   *)
           ZONE  CDECDCDE5664CDECDDCE4700255
          DIGIT  54213198C00D542132580E013CD
                 1...5...10....*...20....*..
```

Abend-AID for IMS consists of modules that locate and interpret IMS control blocks. It includes specialized diagnostic modules for IMS-generated user abends. An Abend-AID for IMS report provides you with the following information:

- Extended Error Analysis section to diagnose the cause of most IMS user abends and to provide possible solutions to the problem.
- Contents of the Key Feedback Area in an expanded format.

- Trace of the last six calls to the IMS database for every database PCB in the application program's PSB. All non-blank status codes are displayed with their respective descriptions.
- Display of the current segment from the I/O area.
- Segment Search Arguments (SSA) displayed in an easy-to-read format. The segments used in the most recent IMS call are identified and printed with their applicable names and lengths.
- DFSPRPX0 and DFSECP modules printed in hexadecimal format to assist in the resolution of the abend.
- Database call statistics from the PST accounting area, for reference when debugging loops and reviewing I/O.
- IMS environment-specific information:
  - Type of IMS environment
  - IMS release number
  - Name of application program and language
  - PSB Name
  - Location of last IMS call in the program
  - Terminal PCBs
  - Alternate terminal PCBs
  - Database PCBs with index PCBs.
- From the IMS PSB:
  - Program name
  - Programming language
- From other control blocks:
  - Release of IMS
  - Region type
  - PSB Name
- When it is possible to determine the last IMS call:
  - Program name and the displacement into the program from which the call was made.
  - PCB used in this call is printed as the current PCB.
- For the current PCB:
  - Key feedback area
  - Database PCBs, the SSA (segment search argument) used in this call
  - Segments returned in the I/O area.
- Information for database PCBs:
  - PCB address, name, number
  - Database name
  - Segment level
  - Status code
  - Process options
  - Segment name last retrieved
  - Key length
  - Number of segments
  - Call trace from the JCB
  - Key feedback area.
- For terminal PCBs:
  - PCB address
  - Logical terminal name
  - Status code from the last call using this PCB
  - JCB call trace for this PCB. Up to the last six calls to the database are given.

File-AID Access

From the IMS section you can directly access an IMS database through File-AID/IMS. Edit and Browse commands in the report take you into File-AID edit and browse modes within the database. A current version of File-AID/IMS must be available on the operating system.

Abend-AID for IDMS

Abend-AID for IDMS captures all data prior to rollback, in contrast to the normal dump, which can display data only after rollback is completed. Thus a true analysis of the data is provided at the time of abend.

You can obtain the following information from an Abend-AID for IDMS report:

- A specialized Error Analysis section that includes:
  - The text and location of the IDMS command in error
  - The particular error status code
  - An error message.
- An IDMS Command Trace section that assists in tracking the program flow of the last ten calls. This section identifies:
  - Program reference statement of each call
  - Complete text of the database command
  - Status code of each call
  - Database key cross-reference with the time and actual database (DB) key in both hexadecimal and decimal.
- A Currency Table that provides references for current:
  - Record
  - Set
  - Area.
- Record information that includes:
  - Current and previous record contents for all record types
  - Working-storage record areas when they differ from current records.
- Database information that includes:
  - Number of commits
  - Most recent commit
  - Database keys in decimal
  - Database statistics section.
- IDMS environmental data that includes:
  - Subschema
  - Version number
  - Region type
  - IDMS release number
  - Central version system ID.

Figure 7-20 shows a sample Abend-AID for IDMS Error Analysis Section for COBOL XLS.

Figure 7-20. Abend-AID for IDMS Error Analysis Section - COBOL XLS

```
    An IDMS U2222 abend occurred following a database call during execution
    of program IDMSCANC.

    ****************************************
    *          Analysis of Error           *
    ****************************************

    CA-IDMS has returned a 0307 error status for the command

    OBTAIN NEXT JOB WITHIN AREA ORG-DEMO-REGION

    This command corresponds to the following statement:

000369                      CALL 'IDMS' USING SUBSCHEMA-CTRL
000370                                        IDBMSCOM (11)
000371                                        SR440
000372                                        ORG-DEMO-REGION
000373                                        IDBMSCOM (43).

    This statement is contained in paragraph "0530-GETJOB" of program
    CMHT1205.

    The end of AREA 'ORG-DEMO-REGION' was reached or the AREA was empty.
    Check the database trace table to see if the most recently retrieved
    record was the last occurrence or the AREA was empty.
```

Figure 7-21 shows a sample Abend-AID for IDMS Error Analysis Section for COBOL Basic Support. This section reports on the status code of the call directly preceding the ROLLBACK. Usually, this call is the failing database call. If you have made additional database calls between the failing database call and the ROLLBACK, Abend-AID for IDMS still reports on the last call preceding ROLLBACK.

Figure 7-21. Abend-AID for IDMS Error Analysis Section - COBOL Basic Support

```
    An IDMS U2222 abend occurred following a database call during execution
    of program IDMSCANC.

    ****************************************
    *          Analysis of Error           *
    ****************************************

    CA-IDMS has returned a 0307 error status for the command
    OBTAIN NEXT EMPLOYEE WITHIN SET EMP-NAME-NDX
    This command is DML sequence number 0007 in program 'COB0307'.

    The end of SET 'EMP-NAME-NDX' was reached or the SET was empty. Check
    the database trace table to see if the most recently retrieved record
    was the last occurrence or the SET was empty.

    See the CA-IDMS environment data below for further details.

                CA-IDMS is a registered trademark of
                Computer Associates International, Inc.
```

Figure 7-22 shows an example of Abend-AID for IDMS call tracing.

Figure 7-22. Abend-AID for IDMS Command Trace

```
                          Database Command Trace
                          ----------------------
   Program
   Reference  Status  Database Command
   ---------  ------  ----------------
   COB0307
   DML
      7       0000    OBTAIN NEXT EMPLOYEE WITHIN SET EMP-NAME-NDX
      7       0000    OBTAIN NEXT EMPLOYEE WITHIN SET EMP-NAME-NDX
      7       0000    OBTAIN NEXT EMPLOYEE WITHIN SET EMP-NAME-NDX
      7       0000    OBTAIN NEXT EMPLOYEE WITHIN SET EMP-NAME-NDX
      7       0000    OBTAIN NEXT EMPLOYEE WITHIN SET EMP-NAME-NDX
      7       0000    OBTAIN NEXT EMPLOYEE WITHIN SET EMP-NAME-NDX
      7       0000    OBTAIN NEXT EMPLOYEE WITHIN SET EMP-NAME-NDX
      7       0000    OBTAIN NEXT EMPLOYEE WITHIN SET EMP-NAME-NDX
      7       0307    OBTAIN NEXT EMPLOYEE WITHIN SET EMP-NAME-NDX
      9       0000    ROLLBACK

  Status Legend
  -------------
  0000   NORMAL RETURN
  0307   END OF SET
```

Figure 7-23 shows an example of Abend-AID for IDMS currency information. Currency tables list the database keys translated to page and line, program reference (DML offset, sequence number or statement number), and usage mode for each area. At a glance, you can see what currency has been established.

Figure 7-23. Abend-AID for IDMS Currency Information

```
                                Current Records
                                ---------------
                                        DB-Key   (Decimal)
          Record Name         DB-Key       Page       Line
          -----------        --------   ----------   ------
          EMPLOYEE           01252A01      75,050        1
          DEPARTMENT         No Currency
          STRUCTURE          No Currency
            . . .

                                Current Sets
                                ------------
   Set Name          Record Name         Program    DB-Key       Page       Line
                                         Reference
   ---------         ------------        ---------  ------    ----------   -----
                                         COB0307
                                         DML
   EMP-NAME-NDX                              7      0124FA01      75,002       1
   COVERAGE-CLAIMS                       None       No Currency
   DEPT-EMPLOYEE     EMPLOYEE            None       01252A01      75,050       1
   EMP-COVERAGE      EMPLOYEE            None       01252A01      75,050       1
     . . .

                                Current Areas
                                -------------
Area Name        Record Name      Usage Mode     Program     DB-Key       Page        Line
                                                 Reference
---------        -----------      ----------     -------     ------    ----------    ------
EMP-DEMO-REGION                   RETRIEVAL      None        0124FA01      75,002        1
INS-DEMO-REGION                   RETRIEVAL      None        No Currency
ORG-DEMO-REGION                                  None        No Currency
```

Figure 7-24 shows an example of the current and previous record areas. Contents for all records are shown. Working-storage record contents are displayed if they differ from the current record area.

Figure 7-24. Abend-AID for IDMS Current and Previous Record Areas

```
                                        Current contents for record EMPLOYEE
                                        Record Length    116
                                        DB-Key           01252A01
                                        DB-Key Page      75,050
                                        DB-Key Line      1

0007295C   CHAR  0476BETSY     ZEDI           34 VALE AVE         SOUTHBORO      MA03145    617431990905010004560760 2
            ZONE  FFFFCCEEE44444ECCC44444444444FF4ECDC4CEC444444444EDEECCDDD444444DCFFFFF4444FFFFFFFFFFFFFFFFFFFFFFFF
           DIGIT  0476253280000095490000000000340513501550000000002643826960000004103145000061743199090501000456076 2
                  1...5...10....*...20....*...30....*...40....*...50....*...60....*...70....*...80....*...90....*..100
000729C0   CHAR  23000000401229
            ZONE  FFFFFFFFFFFFF00
           DIGIT  2300000040122900
                101...5...10....*.

                                        Working storage is equal to current record area

                                        Previous contents for record EMPLOYEE
                                        DB-Key           01251F01
                                        DB-Key Page      75,039
                                        DB-Key Line      1

000728E8   CHAR  0472ROBBY     WILDER         4567 E. GROWTH ST   SOUTHBORO      MA03145    617431770901038779010790 7
            ZONE  FFFFDDCCE44444ECDCCD444444444FFFF4C44CDDEEC4EE444EDEECCDDD444444DCFFFFF4444FFFFFFFFFFFFFFFFFFFFFFFF
           DIGIT  0472962280000069345900000000004567 05B07966380230002643826960000004103145000061743177090103877901079 7
                  1...5...10....*...20....*...30....*...40....*...50....*...60....*...70....*...80....*...90....*..100
0007294C   CHAR  16000000550304
            ZONE  FFFFFFFFFFFFF00
           DIGIT  1600000055030400
                101...5...10....*.
```

With SNAP-AID

Abend-AID for IDMS is available for CA-IDMS Release 12.01 and more current, and for programs that are run in the central version or local mode. It automatically provides IDMS data for non-IDMS abends such as an S0C7 system abend.

With SNAP-AID, you can dynamically obtain complete IDMS information during program execution. If you choose to continue program processing when not using the auto-status precompile option, call the module SNAPAID in your program to produce a complete Abend-AID XLS report with IDMS-specific information. Refer to **Chapter 8, "Using SNAP-AID"** for additional information.

Note: Abend-AID XLS incurs a minimum overhead of 16K for each bind run unit issued. This storage remains allocated until step termination. If your application program issues a large number of bind run units (hundreds, for example), an S106-C abend may occur. Refer to the "Free Abend-AID for IDMS Work Areas" global option in Appendix C, "Global Options" of the *Abend-AID XLS Installation and Customization Guide*.

You can set this option to **Y** if a potential for S106 (out of memory) abends exists with certain application programs. The //ABNLIFRE DD statement is also available for individual application programs. If specified in the execution JCL, this statement causes the work areas that are acquired by Abend-AID for IDMS for each bind run unit to be freed when a finish is issued.

Chapter 8.
Using SNAP-AID

This chapter describes how to use Abend-AID's SNAP-AID facility. You can use SNAP-AID to produce Abend-AID output without terminating the program. The SNAP-AID report includes working storage and file information. SNAP-AID is invoked by a simple call to the module SNAPAID from COBOL, PL/I, Assembler, or FORTRAN programs. SNAP-AID can help in resolving logic problems or when working with a DB2 or IDMS database management system.

Invoking SNAP-AID

Note: Do not call SNAP-AID from a program running in access register mode. Exit access register mode before calling SNAP-AID, then return to access register mode when execution of your program resumes.

To invoke SNAP-AID with a DD specified, perform the following steps:

1. Code a call to the module SNAPAID or SNPAID for FORTRAN. Figure 8-1 shows a SNAPAID call from a COBOL program.

Figure 8-1. SNAPAID Call in a COBOL Program

```
PROCEDURE DIVISION.

(OPEN FILES)

PERFORM READ-RECORD THRU READ-RECORD-EXIT
     UNTIL END-OF-FILE.

READ-RECORD.

      READ INPUT-RECORD INTO WS-RECORD
            AT END MOVE 'Y' TO EOF-SWITCH
                  GO TO READ-RECORD-EXIT.
      ADD 1 TO RECORD-CTR.
      IF WS-EMP-NUMBER IS NOT NUMERIC
             CALL 'SNAPAID'
             GO TO READ-RECORD.
      PERFORM FORMAT-REPORT-RECORD-ROUTINE.
      PERFORM WRITE-REPORT-RECORD-ROUTINE.

READ-RECORD-EXIT.
      EXIT
```

Figure 8-2 on page 8-2 shows SNAPAID calls from a PL/I program.

Figure 8-2. SNAPAID Calls in a PL/I Program

```
 MAIN3: PROC OPTIONS (MAIN);
  /*****************************************************/
  /* SNAPAID CAN BE INVOKED FROM PL/I MAIN PROGRAMS,  */
  /* AND INTERNAL OR EXTERNAL SUBROUTINES:            */
  /*****************************************************/
           .
           .
  /*****************************************************/
  /* DECLARE THE SNAPAID SUBROUTINE AS FOLLOWS:       */
  /*****************************************************/
  DCL SNAPAID EXTERNAL ENTRY
     OPTIONS (ASSEMBLER, RETCODE);
           .
           .
  /*****************************************************/
  /* MAIN PROGRAM BEGINS HERE:                         */
  /*****************************************************/
           .
           .
  SELECT (COLOR);
     WHEN ('GREEN', 'BLUE ', 'WHITE') COLOR_ALL = '1'B;
     WHEN ('NAVY ', 'RED  ', 'BEIGE') COLOR_ALL = '1'B;
     WHEN ('BRWN ', 'TAUPE')          COLOR_MEN = '1'B;
     WHEN ('YELLO', 'PINK ')          COLOR_WMN = '1'B;

     OTHERWISE CALL SNAPAID; <---------SNAPAID used in
  END;                                  OTHERWISE clause
                                        of SELECT statement

  IF ORDER_SIZE= 'XXL' & ÿCOLOR_WMN
     THEN CALL PROCESS_XXLARGE (COLOR);
     ELSE IF ORDER_SIXE = 'XL' & ÿCOLOR_WMN
        THEN CALL PROCESS_XLARGE (COLOR);
        ELSE IF ORDER_SIZE = 'L'
           THEN CALL PROCESS_LARGE (COLOR);
           ELSE IF ORDER_SIZE = 'M'
              THEN CALL PROCESS_MEDIUM (COLOR);
              ELSE IF ORDER_SIZE = 'S'& ¬COLOR_MEN
                 THEN CALL PROCESS_SMALL (COLOR);
SNAPAID used     ELSE IF ORDER_SIZE = 'P' & ¬COLOR_MEN
in last nested      THEN CALL PROCESS_PETITE (COLOR);
IF statement --------->ELSE CALL SNAPAID;
```

2. Compile and link-edit the program using one of the following:
   - The DYNAM compiler option, which is available for COBOL
   - A //SYSLIB DD statement for the link edit that includes the Abend-AID load library:

     ```
     //SYSLIB DD DSN=SYS1.ABENDAID,DISP=SHR
     ```

   - A linkage editor INCLUDE statement for the SNAPAID subroutine module:

     ```
     //SYSIN DD *
       INCLUDE SYSLIB(SNAPAID)
     ```

   SNAPAID is a reentrant module. In MVS systems, the module SNAPAID is linked in the addressing mode AMODE=ANY and residency mode RMODE=24. In order to include SNAPAID in a load module with residency mode RMODE=ANY, do one of the following:
   - Code the linkage editor statement MODE with the RMODE value.
   - Specify the RMODE value as a parameter in the PARM field of the EXEC statement passing it to the linkage editor.
3. Add an //ABENDAID DD statement to the execution JCL.

The SNAP-AID report is written to an output destination chosen in the same manner as is done for a normal Abend-AID XLS report. The only exception to this is that a SNAP-AID report cannot be written to the SYSUDUMP and SYABEND IBM dump DDs. Normally, the

SNAP-AID report is written using the site default report shared directory named by CWROUTE. However, this can be overridden as follows:

- //ABNLTERM DD, which overrides
- CWEXIT02, which overrides
- CWJOBTAB, which overrides
- a CWROUTE routing group specification, which overrides
- the CWROUTE site default report shared directory specification.

To write the SNAP-AID report to the //ABENDAID DD, use CWEXIT02, CWJOBTAB, or a CWROUTE routing group to designate that the report be written to SYSOUT. Refer to the *Abend-AID XLS Installation and Customization Guide*, Chapter 5, "Customization" for more information about routing report output.

SNAP-AID Execution

SNAP-AID performs identically to Abend-AID with the following exceptions:

- An //ABENDAID DD statement *must* be present to produce SNAP-AID output. If one is not present, it may be dynamically allocated.
- SNAP-AID reports the pseudo abend code SNAP.
- The SNAP-AID report is entitled "Abend-AID Snapshot."
- The system produces no IBM dump.
- The //ABNLIGNR DD statement has no effect on SNAP-AID and does not suppress SNAP-AID output.
- The //ABNLIGNS DD statement can be used to suppress SNAP-AID output.

Using SNAP-AID and XPEDITER/TSO

To take full advantage of the Abend-AID SNAP-AID feature while using XPEDITER/TSO, the ABENDAID DD must be allocated to the test session prior to the SNAP-AID call. An Abend-AID report can also be generated any time during the interactive session by issuing the AA SNAP command. If the AA SNAP command is used, the ABENDAID DD will be dynamically allocated by XPEDITER/TSO.

SNAP-AID Return Codes

Return codes from SNAP-AID are:

0:

OK. SNAP-AID is complete.

4:

SNAP-AID cannot locate the required Abend-AID modules, or a compatible Abend-AID hook is not installed. SNAP-AID is bypassed.

8:

No //ABENDAID DD statement is available, or the //ABENDAID DD statement DCB attributes are invalid.

12:

There is not enough memory to allow SNAP-AID to run. SNAP-AID is bypassed.

16:

SNAP-AID execution abnormally terminated.

20:

A subsequent call has been made to SNAP-AID, but SNAP-AID has been disabled for one of the reasons listed above.

SNAP-AID Parameter List

The SNAP-AID parameter list lets you do the following:

- Customize your SNAP-AID report by selecting or suppressing specified report sections. These report sections include:
  - Location in program of the call to SNAP-AID
  - Register contents and PSWs
  - Trace of program flow and program attributes
  - Program Storage
  - File information - data management control blocks
  - IMS diagnostic information and PCBs
  - IDMS diagnostic and status information
  - DB2 diagnostic and status information
  - Abend-AID for WebSphere MQ diagnostic and status information
  - Current COBOL sort information.
- Identify individual SNAP-AID reports in the report dataset directory when using Compuware/VF.
- Cause dynamic allocation of the //ABENDAID DD. The creation of the dynamic allocation request can receive input from various sources and employs a hierarchy in deciding which source to create the dynamic allocation parameters from:
  a. Data passed in the parameter list takes precedence over all other sources.
  b. For optional parameter values not passed, model information may be extracted from one of the following sources:
     - An existing //CEEDUMP DD
     - An existing //SYSUDUMP DD
     - An existing //PLIDUMP DD
     - An existing //PL1DUMP DD.

     If the //CEEDUMP DD was dynamically allocated, it will not be considered eligible for a model. If more than one of the above DDs exists, the first one found in the above order will be used as a model.
  c. If no existing //CEEDUMP, //SYSUDUMP, //PLIDUMP, or //PL1DUMP DD is found, the default values established by the Abend-AID installer will be used.
  d. If no existing //SYSUDUMP DD is found, the default values established by the Abend-AID installer will be used.
  e. If the Abend-AID installer did not establish one or more defaults, the Abend-AID defaults will be used.
- Display storage ranges between selected addresses.

To code a SNAP-AID parameter list, define the parameter interface in your program with the following fields:

- **Identification:** This six-byte field must be coded as SNPRML.
- **Action flag:** This field is used to select the action (display or suppress) that SNAPAID applies to the sections specified. Valid values are:

  - 0 (default): Display all report sections
  - 1: Display specified report sections
  - 2: Suppress specified report sections.
- **Section selection flags:** When the action flag is set to 1 or 2, use the section selection flag(s) to identify the report section(s) to be displayed or suppressed. Valid values are:
  - 0 (default): Do not apply the action specified by the action flag.
  - 1: Apply the action specified by the action flag.
- **Parameter list version:** When the version is set to 1, the dynamic allocation parameters are recognized. Otherwise, they are not recognized.
- **Comment:** This ten-character field is specified in the program to identify the SNAPAID call that generated the report. The content of the comment field is displayed in the DESC field on the report dataset directory.
- **Dynamic allocation request:** When the request is set to D, dynamic allocation of the //ABENDAID DD is requested. When the request is set to F, dynamic allocation and FREE=CLOSE are requested. Otherwise, dynamic allocation is not requested.

  Note: If an //ABNLTERM DD is allocated by your JCL or dynamically allocated because of a global request, then the SNAP-AID report will be routed to the //ABNLTERM allocation and not to the //ABENDAID allocation.

- **Output class:** When the class is set to any valid JES class, that class is used for the allocation. A class of * denotes usage of the MSGCLASS value. Specifying an invalid class or no class causes the model or default class to be used.
- **Hold the output:** When the optional hold parameter is set to Y, the output will be placed in the hold queue by JES. When hold is set to N, the output will be directed to the output queue by JES. Otherwise, whether to hold output or not will be determined by the model or default hold specification.
- **Destination:** This optional eight-character field specifies the node to which you want to direct the output. Consult the DD DEST section of your JCL reference manual for more information on nodes. If you leave this field blank, the destination will be determined by the model or default destination value.
- **User ID:** This optional eight-character field specifies the user ID at the node destination where that you want the output routed. If you leave this field blank, the user ID will be determined by the model or default user ID value.
- **Output writer name:** This optional eight-character field specifies a custom output writer to be used to process the output. Note that the use of this parameter is mutually exclusive with user ID. If both are specified, user ID will be used. Output writer names are site-dependent.
- **User form name:** This optional four-character field specifies the form name to be used for the output. If you leave this field blank, the form name will be determined by the model or default form name value. User form names are site-dependent.

  Note: The Abend-AID installation library contains parameter list interface layouts: member PRMICBL for COBOL, member PRMIPLI for PL/I, and member PRMIASM for Assembler.

- You can use two additional parameters to specify a range of storage to be displayed in the Program Storage section. The field name specified as the second parameter defines the beginning storage address. The third name specified as the third parameter defines the ending storage address. Subsequent pairs of parameters are processed as additional ranges of storage to be displayed. If these parameters are used, the specified ranges of storage replace the Program Storage section.

The following are examples of SNAPAID calls in the Procedure Division to produce a SNAP-AID report:

Example 1: Without using a parameter list:

- COBOL:

```
CALL 'SNAPAID'.
```

- Assembler:

```
         LA   R1,0            Indicate 'no parameter list'
         CALL SNAPAID
```

Example 2: Using a parameter list:

- COBOL:

```
CALL 'SNAPAID' USING PARM-INTERFACE.
```

- PL/I:

```
CALL SNAPAID (PARM_INTERFACE)
```

- Assembler:

```
         OI   ADDR0,X'80'      Set end-of-list bit
         LA   R1,PARMLIST      Point to parameter list
         B    AROUND_LIST
PARMLIST DS   0F
ADDR0    DC   A(SNPMITR)       Address of SNAPAID parameters
AROUND_LIST EQU *
         CALL SNAPAID
```

Example 3: Using a parameter list specifying a range of addresses:

- COBOL:

```
CALL 'SNAPAID' USING PARM-INTERFACE
     WS-RANGE1-BEG WS-RANGE1-END.
```

- PL/I:

```
CALL SNAPAID (PARM_INTERFACE,
RANGE1_BEG,RANGE1_END);
```

- Assembler:

```
         OI   ADDR2,X'80'      Set end-of-list bit
         LA   R1,PARMLIST      Point to parameter list
         B    AROUND_LIST
PARMLIST DS   0F
ADDR0    DC   A(SNPMITR)       Address of SNAPAID parameters
ADDR1    DC   A(BEGDATA)       Address of storage dump start
ADDR2    DC   A(ENDDATA)       Address of storage dump stop
AROUND_LIST EQU *
         CALL SNAPAID
```

Example 4: Using a parameter list specifying ranges of addresses:

- COBOL:

```
CALL 'SNAPAID' USING PARM-INTERFACE
     WS-RANGE1-BEG WS-RANGE1-END
     WS-RANGE2-BEG WS-RANGE2-END.
```

- PL/I:

```
CALL SNAPAID (PARM_INTERFACE
RANGE1_BEG,RANGE1_END,
RANGE2_BEG,RANGE1_END);
```

- Assembler:

```
         OI   ADDR4,X'80'      Set end-of-list bit
         LA   R1,PARMLIST      Point to parameter list
         B    AROUND_LIST
PARMLIST DS   OF
ADDR0    DC   A(SNPMITR)       Address of SNAPAID parameters
ADDR1    DC   A(BEGDATA)       Address of storage dump start
ADDR2    DC   A(ENDDATA)       Address of storage dump stop
ADDR3    DC   A(BEGDATA#)      Address of storage dump start
ADDR4    DC   A(ENDDATA#)      Address of storage dump stop
AROUND_LIST EQU *
         CALL SNAPAID
```

Note: The Abend-AID installation library contains sample programs that produce SNAP-AID reports using a parameter list: member PRMSCBL for COBOL, member PRMSPLI for PL/I, and member PRMSASM for Assembler.

Figure 8-3. Parameter Interface Defined in a COBOL Program

```
 01  PARM-INTERFACE.
     03  PARM-IDENT             PIC X(6)  VALUE 'SNPRML'.
     03  PARM-ACTION            PIC 9     VALUE ZERO.
         88  DISPLAY-ALL-SECTIONS         VALUE 0.
         88  DISPLAY-SECTION              VALUE 1.
         88  SUPPRESS-SECTION             VALUE 2.
*            0 FULL SNAPAID REPORT (NO CUSTOMIZATION)
*            1 DISPLAY SELECTED SECTIONS
*            2 SUPPRESS SELECTED SECTIONS
     03  PARM-SECTION-SELECTION.
         05  NSI-SECTION        PIC 9     VALUE ZERO.
             88 DO-NOT-APPLY-PARM-ACTION  VALUE 0.
             88 APPLY-PARM-ACTION         VALUE 1.
*               0 DO NOT APPLY PARM-ACTION TO THIS SECTION
*               1 APPLY PARM-ACTION TO THIS SECTION
         05  REGISTERS-PSW      PIC 9     VALUE ZERO.
         05  TRACE-SUMMARY      PIC 9     VALUE ZERO.
         05  PROGRAM-STORAGE    PIC 9     VALUE ZERO.
         05  FILES-SECTION      PIC 9     VALUE ZERO.
         05  IMS-SECTION        PIC 9     VALUE ZERO.
         05  IDMS-SECTION       PIC 9     VALUE ZERO.
         05  DB2-SECTION        PIC 9     VALUE ZERO.
         05  SORT-SECTION       PIC 9     VALUE ZERO.
         05  ECM-SECTION        PIC 9     VALUE ZERO.
         05  PARML-VERSION      PIC 9     VALUE ZERO
*        05  DYNAMICAL ALLOCATION INFORMATION NOT PRESENT IF BLANK
*        05 DYNAMIC ALLOCATION INFORMATION PRESENT
         PARM-COMMENT           PIC 9     VALUE SPACE.
*        DYNALC-REQ             PIC X(10) VALUE SPACE.
        3 88  DON'T-ALLOCATE-ABENDAID     VALUE SPACE.
         88  ALLOCATE-ABENDAID            VALUE 'D'.
         88  ALLOC-AND-FREE-ON-CLOSE      VALUE 'F'.
*              THE ABENDAID DD WILL NOT BE DYNAMICALLY ALLOCATED IF
*              BLANK
*            D DYNAMICALLY ALLOCATE THE ABENDAID DD
*            F DYNAMICALLY ALLOCATE THE ABENDAID DD AND FREE=CLOSE
     03  OUTPUT-CLASS           PIC X     VALUE SPACE.
     03  HOLD-OUTPUT            PIC X     VALUE SPACE.
         88  DONT-HOLD-DEF                VALUE SPACE.
         88  HOLD                         VALUE 'Y'.
         88  DONT-HOLD                    VALUE 'N'.
     03  FILLER                 PIC X     VALUE SPACE.
     03  DESTINATION-PARM       PIC X(8)  VALUE SPACE.
     03  USERID                 PIC X(8)  VALUE SPACE.
     03  OUTPUT-WRITER          PIC X(8)  VALUE SPACE.
     03  USER-FORM              PIC X(4)  VALUE SPACE.
```

Figure 8-4. Parameter Interface Defined in a PL/I Program

```
   DECLARE           1   PARM_INTERFACE,
            2  PARM_IDENT          CHAR(6)  INIT('SNPRML'),
            2  PARM_ACTION         CHAR(1)  INIT('0'),
  /*           = 0 FULL SNAPAID REPORT (NO CUSTOMIZATION)      */
  /*           = 1 DISPLAY SELECTED SECTIONS                   */
  /*           = 2 SUPPRESS SELECTED SECTIONS                  */
            2  PARM_SECTION(11)
                   CHAR(1)  INIT((11)('0'),(1)('1')),
  /*           (1) NSI                                         */
  /*           (2) REGISTERS-PSW                               */
  /*               = 0 DO NOT APPLY PARM_ACTION FOR THIS SECT  */
  /*               = 1 APPLY PARM_ACTION FOR THIS SECTION      */
  /*           (3) TRACE-SUMMARY                               */
  /*           (4) PROGRAM-STORAGE                             */
  /*           (5) FILES                                       */
  /*           (6) IMS-SECTION                                 */
  /*           (7) IDMS-SECTION                                */
  /*           (8) DB2-SECTION                                 */
  /*           (9) SORT-SECTION                                */
  /*         (10) UNUSED                                       */
  /*         (11) PARML-VERSION                                */
  /*           = 1 DYNAMIC ALLOCATION DATA IS PRESENT          */
            2  PARM_COMMENT        CHAR(10)  INIT((10)' '),
            2  DYNALC-REQ          CHAR(1)   INIT('F'),
  /*           = D DYNAMIC ALLOCATION OF THE ABENDAID DD       */
  /*           = F DYNAMICALLY ALLOCATE AND DO FREE=CLOSE      */
            2  OUTPUT-CLASS        CHAR(1)   INIT('A'),
  /*           = X ANY VALID OUTPUT CLASS                      */
            2  HOLD-OUTPUT         CHAR(1)   INIT('Y'),
  /*           = Y HOLD THE OUTPUT                             */
  /*           = N DON'T HOLD THE OUTPUT                       */
            2  FILLER              CHAR(1)   INIT(' '),
            2  DESTINATION         CHAR(8)   INIT('RMT5'),
            2  USERID              CHAR(8)   INIT('MYID'),
            2  OUTPUT-WRITER       CHAR(8)   INIT(' '),
            2  USER-FORM           CHAR(4)   INIT('SPCL');
 DCL SNAPAID EXTERNAL ENTRY OPTIONS(RETCODE,ASSEMBLER);
```

Figure 8-5. Parameter Interface Defined in an Assembler Program

```
SNPRMITR   DS   OH
PARMIDEN   DC   CL6'SNPRML'   +0  CALLER ID
*
PARMACTN   DC   CL1'0'        +6  ACTION FLAG
*  EQUATES TO BE USED IN SETTING PARMACTN (ACTION).
FULLFLG    EQU  C'0'               FULL SNAPAID REPORT
*                                       (NO CUSTOMIZATION)
DISPLYFLG  EQU  C'1'               DISPLAY SELECTED SECTIONS
SUPPRFLG   EQU  C'2'               SUPPRESS SELECTED SECTIONS
*
PARMSECT   EQU  *          SECTION FLAGS - SET TO "0" OR "1"
NSISCT     DC   CL1'0'        +7  NSI
REGSCT     DC   CL1'0'        +8  REGISTERS-PSW
TRACSCT    DC   CL1'0'        +9  TRACE-SUMMARY
PROGSCT    DC   CL1'0'       +10  PROGRAM-STORAGE
FILESCT    DC   CL1'0'       +11  FILES
IMSSCT     DC   CL1'0'       +12  IMS-SECTION
IDMSSCT    DC   CL1'0'       +13  IDMS-SECTION
DB2SCT     DC   CL1'0'       +14  DB2-SECTION
SRTSCT     DC   CL1'0'       +15  SORT-SECTION
ECMSET     DC   CL1'0'       +16  E-COMMERCE-SECTION
PRMLVERS   DC   CL1'1'       +17  DYNAMIC ALLOCATION DATA PRESENT
*  EQUATES TO BE USED IN SETTING SECTION FLAGS ABOVE.
SECTNOFF   EQU  C'0'               DO NOT APPLY DISPLAY/SUPPRESS
*                                       ACTION TO THIS SECTION.
SECTNON    EQU  C'1'               APPLY DISPLAY/SUPPRESS ACTION
*                                       TO THIS SECTION.
PARMCMNT   DC   10CL1' '     +18  COMMENT - APPEARS IN PROGRAMMER
*                                  COLUMN OF REPORT FILE DIRECTORY
DYNASECT   EQU  *          DYNAMIC ALLOCATION SECTION
DYNALC     DC   CL1' '       +28  DYNAMICALLY ALLOC ABENDAID
*                                  AND OPTIONALLY SPECIFY
*                                  FREE=CLOSE
DYNALLOC   EQU  C'D'                DYNAMICALLY ALLOC ABENDAID
FREEQCLS   EQU  C'F'                SPECIFY FREE=CLOSE TOO
OUTPTCLS   DC   CL1' '       +29  SYSOUT CLASS TO BE USED
HOLDOUTP   DC   CL1' '       +30  HOLD THE OUTPUT (Y/N)
UNUSE2     DC   CL1' '       +31  UNUSED
DESTNATN   DC   CL8' '       +32  DESTINATION (LOCAL / REMOTE)
USERID     DC   CL8' '       +40  USER I.D.
OUTPWRTR   DC   CL8' '       +48  OUPUT WRITER NAME
USERFORM   DC   CL4' '       +56  USER FORM NAME
SNPRMLNG   EQU   *-SNPRMITR
           END
```

Suppressing SNAP-AID

SNAP-AID output may be suppressed without removing the call to SNAP-AID from your program. The way you do this depends on how the //ABENDAID DD statement is allocated.

- If the //ABENDAID DD is allocated by a DD statement in the execution JCL, remove the //ABENDAID DD statement from the JCL.
- If the //ABENDAID DD is dynamically allocated because of a request in the SNAP-AID parameter list, add the //ABNLIGNS DD to the execution JCL.

Chapter 9.
Using Abend-AID DD Statements

This chapter contains information about using the Abend-AID XLS DD statements. You can add DD statements to the execution JCL to override specifications in the Abend-AID XLS tables and user exits.

The following types of Abend-AID XLS DD statements can be coded:

- General usage
- IBM dump-related
- Processing control
- Report output
- Storage-related
- Record format
- IDMS-related

The Abend-AID XLS tables and user exits are installed and modified by the Abend-AID XLS installer. The tables and user exits listed below may be installed at your site.

Abend-AID XLS Tables

- Global site options table (CWGLOBAL) created during installation
- Routing criteria (CWROUTE) created during installation
- Job and program selection tables (CWJOBTAB and CWPGMTAB)
- Abend code tables (CWTABS01, CWTABU01, CWTABP01, CWTABL01)
- CSECT inclusion table (CWINCLUD)
- CSECT bypass table (CSECTBYP).

Abend-AID XLS User Exits

- Processing control (CWEXIT01)
- Report routing and control (CWEXIT02)
- Customized storage display (CWEXIT03).

Control Hierarchy

When the DD statements, tables, and user exits are used, there is a hierarchy that determines Abend-AID XLS control. The following control hierarchy within Abend-AID XLS defines the order in which overrides occur:

- JCL DD statements override what is specified in user exits, job and program selection tables, routing criteria, and the global site options table.
- User exits override job and program selection tables, routing criteria, and the global site options table.
- The job selection table overrides what is specified in the program selection table, routing criteria, and the global site options table. The program selection table is ignored when a match is found in the job selection table.
- The program selection table overrides what is specified in the routing criteria and global site options table.

Abend-AID XLS DD Statements

Any of the DD statements listed in this section can be included in the execution JCL to control Abend-AID XLS processing and/or reports. If two mutually exclusive DD statements are present in the JCL, the last DD statement encountered will be in effect. Mutually exclusive DD statements are noted in this chapter.

General Usage

Following is a description of the general usage DD statements:

//SYSUDUMP DD SYSOUT **or** **//SYSABEND DD SYSOUT** **or** **//SYSMDUMP DD SYSOUT**	Any one of these DD statements will invoke Abend-AID XLS. However, users of //SYSMDUMP must use the //ABENDAID DD (see below) to receive Abend-AID XLS output because Abend-AID XLS does not write to //SYSMDUMP. If more than one DD is specified, the last listed DD is used.
//ABENDAID DD SYSOUT	An alternate DD for Abend-AID XLS output. If it is not present in the JCL, it may be dynamically allocated. A //SYSUDUMP or //SYSABEND DD SYSOUT is still required. Abend-AID XLS's output is written to the //ABENDAID DD when using Language Environment support or Abend-AID XLS's abending or non-abending PL/I support. **JCL Override:** When //ABENDAID DD SYSOUT is specified, all output goes to //ABENDAID DD SYSOUT, not to //SYSUDUMP or //SYSABEND.
//ABNLHELP DD DUMMY	Produces the Abend-AID XLS Help pages. These pages go to the same DD specified for the Abend-AID XLS report output. **Table Override:** //ABNLHELP DD overrides the CWGLOBAL parameter HELP=NO, which suppresses the Help pages. **Table Usage:** Help pages can be either displayed or suppressed globally by using the Help parameter in the CWGLOBAL table.
//ABNLIGNR DD DUMMY	Suppresses Abend-AID XLS and allows normal IBM dump processing.
//ABNLIGNS DD DUMMY	Suppresses SNAP-AID and allows normal processing to continue.
//ABNLTERM DD DSN	Specifies an Abend-AID XLS report dataset for Abend-AID XLS output.
//ABNLTERM DD SYSOUT	Routes Abend-AID XLS output to SYSOUT in 72-column format.
//ABNLWIDE DD DUMMY	Specifies 121-column output when the default is 72 columns.

IBM Dump-Related

Following is a description of the IBM-related DD statements:

//ABNLDUMP DD DUMMY Requests printing of the normal IBM dump in addition to the Abend-AID XLS report. Dump output is routed to //SYSUDUMP DD SYSOUT or //SYSABEND DD SYSOUT. It is not routed to the Abend-AID XLS report dataset or to Abend-AID XLS-specific SYSOUTs. Refer to the overrides.

//ABNLFMTD DD DUMMY Requests printing of only the formatted control blocks part of the IBM dump as defined by your installation. Refer to the overrides.

//ABNLNODP DD DUMMY Suppresses printing of the normal IBM dump.

Mutually Exclusive: //ABNLDUMP and //ABNLFMTD are mutually exclusive of //ABNLNODP.

Overrides

User Exit Override: //ABNLDUMP, //ABNLFMTD, or //ABNLNODP DD takes precedence over printing or suppressing the IBM dump as specified in the CWEXIT01 abend-handling user exit and the abend-handling tables below.

Method of Hierarchy: The abend-handling user exit and tables are listed by precedence; that is, each one listed takes priority over the ones below.

- CWEXIT01
- CWJOBTAB
- CWPGMTAB
- CWTABS01, CWTABU01, CWTABP01, CWTABL01.

Processing Control

Use the following DD statements to control Abend-AID XLS processing.

Enabling Abend-AID XLS

Following is a description of the ABNLENAB DD statement:

//ABNLENAB DD DUMMY Enables Abend-AID XLS processing when it has been disabled in the CWGLOBAL table.

If //ABNLENAB DD is present in the JCL, it can be overridden, and Abend-AID XLS processing will be ignored. See “Bypassing Abend-AID XLS”.

Table Override: Use //ABNLENAB DD to override the “Disable Abend-AID XLS Processing” global options, which disables Abend-AID XLS processing. See Appendix C, “Global Options”.

Bypassing Abend-AID XLS

After Abend-AID XLS processing is active, it can be bypassed in the following ways:

- Include the //ABNLIGNR DD DUMMY statement to ignore Abend-AID XLS processing for a particular job step.

- Include the //ABNLIGNS DD DUMMY statement to ignore SNAP-AID processing for a particular job step.
- Use the CWEXIT01 processing control exit to ignore Abend-AID XLS processing for a specific abend code or job name.
- Use the CWEXIT02 user exit to bypass Abend-AID XLS processing.
- Use the CWJOBTAB selection table to bypass Abend-AID XLS processing based on a specific job name or job name prefix. Use the CWPGMTAB selection table to bypass Abend-AID XLS processing based on a specific program name or program name prefix.

Report Output

This section discusses how to control where Abend-AID XLS report output is routed. Abend-AID XLS output can be routed to one of the three following destinations:

- Abend-AID XLS report shared directory
- Abend-AID XLS report database attached to a shared directory
- Abend-AID XLS report file
- Sequential file
- SYSOUT.

Report Datasets - Shared Directories, Databases, and "DDIO" Files

Abend-AID XLS uses the Compuware Viewing Facility (Compuware/VF) and the features of an ISPF/PDF environment to provide interactive viewing of reports and source listings.

Abend-AID XLS report datasets, or specifically, report shared directories and the report databases attached to them, as well as DDIO report files, must be preallocated and preformatted. Refer to Chapter 2, "Allocating and Formatting DDIO Files", in the *Compuware Shared Services User/Reference Guide* for information on defining, allocating, and formatting report datasets.

Abend-AID XLS reports can be routed to the site default shared directory or to selected report datasets in any of the following ways:

- Specify a single default Abend-AID XLS report shared directory name in the CWROUTE default set during installation.
- Specify an Abend-AID XLS report dataset name in the output processing user exit, CWEXIT02.
- Use the //ABNLTERM DD DSN to specify an Abend-AID XLS report dataset name. For example, insert the card //ABNLTERM DD DSN=MY.ABEND.REPORTS to route Abend-AID XLS reports to that report dataset.
- CWJOBTAB

Note: //ABNLTERM DD DSN and //ABNLTERM DD SYSOUT are mutually exclusive.

The following overrides occur:

- The //ABNLTERM DD statement overrides what is specified in CWEXIT02, CWJOBTAB, and CWROUTE.
- CWEXIT02 overrides what is specified in CWJOBTAB and CWROUTE.
- CWJOBTAB overrides what is specified in CWROUTE.

Note: SNAP-AID output is also routed to the Abend-AID XLS report dataset. IBM dumps are *not* routed to the Abend-AID XLS report dataset.

Defining a Sequential File

Use the //ABNLTERM DD DSN statement in your JCL to route the Abend-AID XLS output to a sequential file. The DCB attributes must be RECFM=VBA and LRECL=125. Otherwise, Abend-AID XLS issues the error message ABONL-2. If Abend-AID XLS is unable to open this file, one of the following DD statements is used to route the Abend-AID XLS output:

- //SYSUDUMP DD
- //SYSABEND DD
- //ABENDAID DD.

User Exit Override: The //ABNLTERM DD DSN overrides the Abend-AID XLS file name specified in the output processing user exit, CWEXIT02.

Table Override: The //ABNLTERM DD DSN overrides the Abend-AID XLS shared directory from CWROUTE or CWJOBTAB.

Mutually Exclusive: //ABNLTERM DD DSN and //ABNLTERM DD SYSOUT are mutually exclusive.

Routing Output to SYSOUT

Use the //ABNLTERM DD SYSOUT statement in your JCL to route the Abend-AID XLS output to SYSOUT. For example, an //ABNLTERM DD SYSOUT=* card in your JCL routes the Abend-AID XLS report to the MSGCLASS destination as coded on the JOB card.

Specifying Narrow Output

Use the //ABNLTERM DD SYSOUT statement to route output in narrow (72-column) format.

Specifying Wide Output

Use the //ABNLWIDE DD DUMMY statement to specify 121-column output when the default report width has been changed to 72 columns.

User Exit Override - CWEXIT02: The //ABNLTERM DD DSN overrides the Abend-AID XLS file name specified in the output processing user exit, CWEXIT02.

Table Override: The //ABNLWIDE DD statement overrides the following tables:

- CWJOBTAB
- CWPGMTAB
- CWGLOBAL.

Note: Use the //ABNLWIDE DD DUMMY statement with the //ABNLTERM DD SYSOUT statement to cause wide (121-column) output to be written to the //ABNLTERM SYSOUT.

Controlling Report Format

Viewing or printing Abend-AID XLS reports can be done in a narrow (72-column) or wide (121-column) format when routed to SYSOUT or a sequential file. The report format in the Abend-AID XLS report dataset cannot be altered. Refer to Table 9-1 to determine the column width of the Abend-AID XLS report.

Table 9-1. ABNLTERM Decision Table

CWGLOBAL, CWJOBTAB, or CWPGMTAB (ONLINE=)	//ABNLTERM	FORMAT
N	N	121 on //SYSUDUMP
N	Y	72 on //ABNLTERM

Table 9-1. ABNLTERM Decision Table

Y	N	72 on //SYSUDUMP
Y	Y	72 on //ABNLTERM

Storage-Related

Use these DD statements to control displaying program working storage and control block information in the Abend-AID XLS report. Mutually exclusive DD statements are noted.

Overrides

Use the following DD statements to override the specific program working storage and control block setting(s) made in these tables:

Table Override:

- CWGLOBAL
- CWTABS01
- CWTABU01
- CWTABP01
- CWTABL01.

//ABNLALL DD DUMMY Requests formatting of all program storage and data management control blocks.

Mutually Exclusive: //ABNLALL DD and //ABNLNONE DD

//ABNLNONE DD DUMMY Suppresses formatting of all program storage and data management control blocks.

Mutually Exclusive: //ABNLNONE DD and //ABNLALL DD

//ABNLPCBS DD DUMMY Requests displaying of the Abend-AID XLS data management control blocks section.

Mutually Exclusive: //ABNLPCBS DD and //ABNLNCBS DD

//ABNLNCBS DD DUMMY Suppresses displaying of the Abend-AID XLS data management control blocks section.

Mutually Exclusive: //ABNLNCBS DD and //ABNLPCBS DD.

//ABNLWSPT DD DUMMY Requests displaying of the Abend-AID XLS Program Storage section.

Mutually Exclusive: //ABNLWSPT DD and //ABNLNWSP DD

//ABNLNWSP DD DUMMY Suppresses displaying of the Abend-AID XLS Program Storage section.

Mutually Exclusive: //ABNLNWSP DD and //ABNLWSPT DD

//ABNLINCL DD DUMMY Causes all active load modules to be considered for CSECT inclusion when the CWINCLUD table is present. Refer to Chapter 6, "Customizing Abend-AID XLS," in the *Abend-AID XLS Installation and Customization Guide* for additional information.

These are the effects on the Program Storage section:

- Without CWINCLUD, the report shows all Assembler CSECTs on the calling chain.
- With CWINCLUD only, Assembler CSECTs in load modules on the calling chain are eligible for inclusion in the report.
- With CWINCLUD and ABNLINCL, Assembler CSECTs in all currently loaded modules of the abending task are eligible for inclusion in the report.

Record Format

Following is a description of the ABNLSPRT DD statement:

//ABNLSPRT DD DUMMY Displays the entire record in the file summary section of the Abend-AID XLS report when the default has been changed by the "Display the Entire Current Record" CWGLOBAL option to be "NO".

Table Override: CWGLOBAL

IDMS-Related

Following is a description of the IDMS-related DD statements:

//ABNLIGNI DD DUMMY Bypasses only the Abend-AID for IDMS reporting facility.

Note: This does not affect any other Abend-AID XLS report processing, including SNAP-AID.

//ABNLIDTR DD SYSOUT Used by Abend-AID for IDMS to produce a dynamic trace output. Traces and displays all calls made to the IDMS database.

//ABNLIFRE DD DUMMY Causes the work areas that are acquired by Abend-AID for IDMS for each bind run unit to be freed when a finish is issued.

Chapter 10.
Understanding Language Processors

This chapter summarizes the Compuware language processor information presented in the Compuware Shared Services (CSS) user manuals. If you would like detailed information, refer to the applicable chapters of the CSS manuals:

- *Enterprise Common Components Installation & Customization Guide:* Overview and installation chapters.
- *Compuware Shared Services User/Reference Guide:* CSS Utilities and applicable language processor chapters.

Use of Language Processors

The Compuware language processors enable XLS by copying source code and processing information from the compiler listing of a program into a Compuware source listing dataset. This makes XLS source information available for an Abend-AID XLS report. Compuware provides a separate language processor for COBOL, PL/I, and Assembler (HLASM) languages.

The language processors are part of Compuware Shared Services (CSS), which is a set of components that are used by multiple Compuware products. The same Compuware language processor and source listing dataset that enable XLS for Abend-AID XLS may also be used by other Compuware diagnostic and testing products.

Operating Modes

Compuware language processors have two different operating modes and several output and processing options. These are typically set by the installer during Abend-AID XLS and CSS installation but may be changed after installation.

The language processor on your system may be operated in either preprocessing mode or postprocessing mode. The names indicate whether language processing is initiated before or after the compile step. Preprocessing includes both compiling and postprocessing. Postprocessing can also run as a separate step.

Preprocessing and postprocessing have the following differences.
Preprocessing:

- Collects from SYSIN and SYSLIB any source code suppressed with COPY SUPPRESS (COBOL), %NOPRINT (PL/I), or PRINT OFF/NOGEN (Assembler).
- Invokes compiling and postprocessing.
- Provides any compiler options needed for postprocessing.
- Requires minor set-up modifications to the compile step.

Postprocessing:

- Requires setting up a separate step after the compile step.
- May require setup modifications to the compiler options.

The installation and setup instructions recommend preprocessing when possible, except when handling compiler listings in a machine-readable format. If a question exists, apply these guidelines:

- Preprocessing — advisable in development or testing environments, when the program contains suppressed source code, or when the source listing will be accessed by XPEDITER/TSO.
- Postprocessing — advisable in a production environment new to XLS or when running a program that has not had a compiler listing in the source listing dataset.

Processing Methods

A language processor is used most efficiently when it is set up to copy a compiler listing to the source listing dataset whenever a program is compiled. This approach ensures the availability of online XLS in the Abend-AID XLS report whenever an abend occurs. Because postprocessing is executed as a separate step, you can also post-compile a listing into the source listing dataset after an abend occurs or postprocess an existing machine-readable compiler listing into the source listing dataset.

Test programs:

- Preprocessing: Run the preprocessor and link-edit the program.
- Postprocessing: Compile the program, run the postprocessor, and link-edit the program.

Production programs:

- Preprocessing: Run the preprocessor to recompile the program.
- Postprocessing: Recompile the program and run the postprocessor.

Machine-readable compiler listings:

- Postprocessing: Write the compiler listing to a sequential dataset using an appropriate utility program, then run postprocessing with the dataset as input. You can postprocess machine-readable compiler listings from most products including:
  - CA-OPTIMIZER (COBOL)
  - CA-PANVALET
  - CHANGE MAN
  - ENDEVOR
  - Software Configuration and Library Manager (SCLM)
  - SYSOUT Archival and Retrieval System (SAR)

Figure 10-1. Preprocessor Compile Process

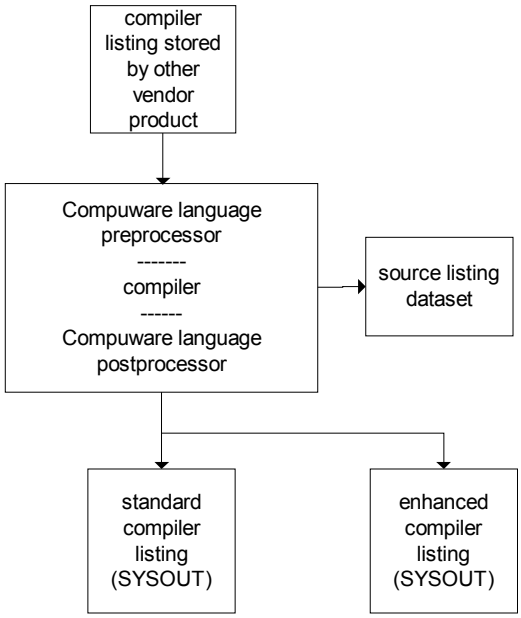


Figure 10-2. Postprocessor As a Step After the Compile

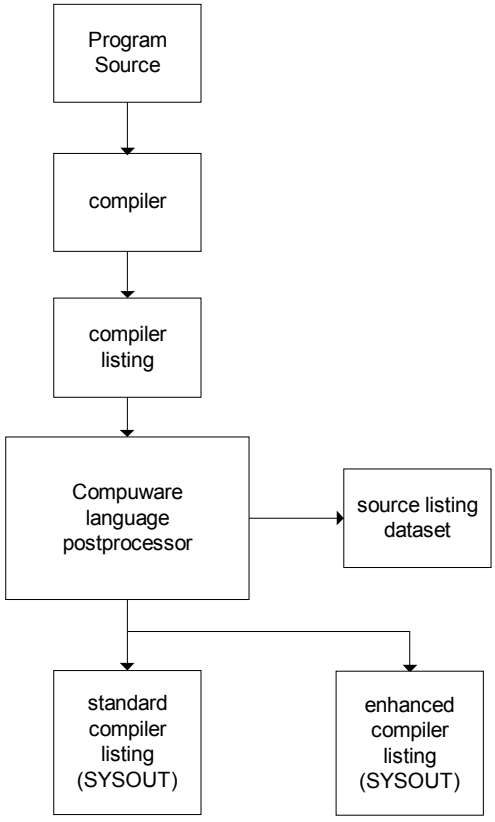
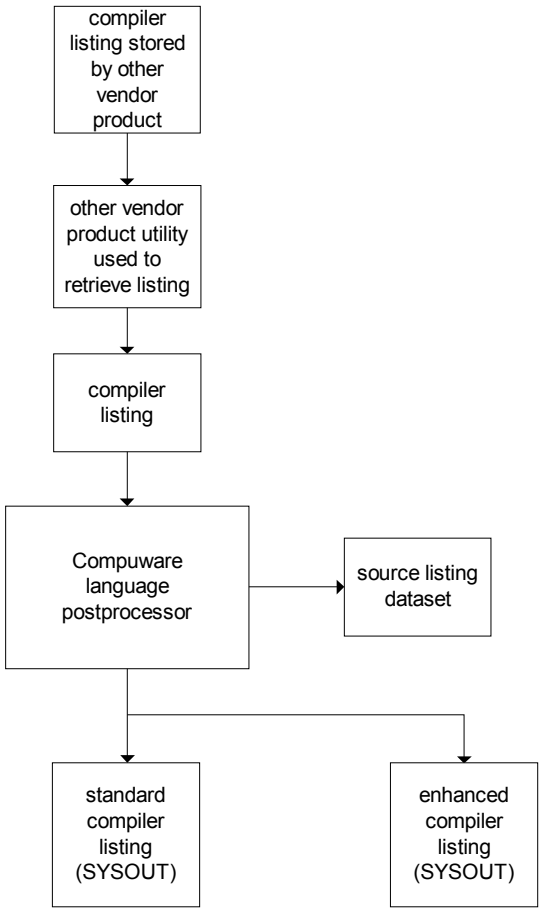


Figure 10-3. Postprocessor Using Stored Compiler Listings



Processing Outputs

Language processor output consists of data copied or derived from the compiler listing. It includes information from the XREF, data maps, and object code sections of the listing. This information is merged into the Abend-AID XLS report when you view it or print it using XLS.

With the COBOL language processor you can specify an enhanced compiler listing. It provides information not found in the standard listing, formatted for easy reference. The additional information is not shown in the Abend-AID XLS report, but it's available for printing or viewing in the source listing dataset. The enhanced listing includes the following information:

- Compiler options, sorted alphabetically
- Merged MAP/DMAP information, BLW, BL, BLL, and displacement into the data division
- OFFSET/CLIST information merged into the procedure division
- Data clauses and their options: Redefines, Occurs, Depending on, File, External, Global
- Decimal format: record lengths, beginning and ending positions of data elements, location and length of items in working storage.

The enhanced listing is not available in PL/I and Assembler XLS.

Output Options

You can use language processor parameters to customize printed and stored compiler listings. A language processor will not copy the compiler listing, or the COBOL enhanced compiler listing, to the source listing dataset unless the applicable output option is set. Compuware's default language processor options do not automatically direct output to the source listing dataset.

Each language processor provides similar outputs that are set during ECC/CSS installation. The COBOL language processor has additional options, including those for the enhanced compiler listing.

Several of the output alternatives are shown below. If you need more information about them, refer to the applicable chapters of the *Enterprise Common Components Installation and Customization Guide* and the *Compuware Shared Services User/Reference Guide*.

Print:

- Enhanced compiler listing (COBOL) *(*Compuware *default)*
- Standard compiler listing
- Both listings (COBOL)
- No listing.

Include in or exclude from output to print: (Compuware default: Include)

- XREF (COBOL, PL/I, ASSEMBLER)
- LIST (COBOL, PL/I)
- OFFSET (COBOL, PL/I)
- DMAP (COBOL)
- Source (COBOL, PL/I, ASSEMBLER)

Store in source listing dataset: (Output to the source listing dataset must be specified)

- Enhanced compiler listing (COBOL)
- Standard compiler listing
- No listing.

Include in or exclude from output to source listing dataset: (Compuware default: Include)

- XREF (COBOL)
- PMAP/LIST (COBOL)
- CLIST/OFFSET (COBOL)

Enhanced compiler listing to source listing dataset or to print:

- Display or do not display compiler warning messages at the front of the listing.
- Embed or do not embed compiler warning messages after the statement in error.
- Print and embed warning messages. *(*Compuware *default)*
- Print error messages in uppercase or mixed-case. *(*Compuware *default: uppercase)*
- In the "From-Thru" section, start offsets from one (positional) or zero (offset). *(*Compuware *default: one)*

Change Language Processing Mode

A Compuware language processor is set up during Abend-AID XLS and ECC/CSS installation. But you can change processing modes for a single processing run, or prepare new report and source listing datasets, with the CSS Utilities.

You can access the CSS Utilities on the Abend-AID XLS Primary Menu. For information about using the utilities, refer to the help selections on the menu, to the Compuware/VF online help, or to the CSS Utilities chapter of the *Compuware Shared Services User/Reference Guide*.

Change Language Processor Options

1. Refer to the applicable language processor chapter of the *Compuware Shared Services User/Reference Guide* to familiarize yourself with available options.

2. Determine whether the options were specified at setup in the program EXEC statement, members or statements CWPPRMO or CXLPCOBB, or elsewhere. Installation instructions recommend CWPPRMO.
3. Change the options as desired.

Chapter 11.
Using Abend-AID in Language Environment

This chapter contains basic information for Abend-AID XLS users who are using Language Environment. In addition to showing Abend-AID's LE support, it answers typical questions about Abend-AID XLS's operation in Language Environment, an IBM run-time environment. If you need general information about Language Environment, refer to the IBM *z/OS Language Environment Concepts Guide*.

Basic Operation

Abend-AID XLS diagnoses errors occurring in programs running in Language Environment (LE). Abend-AID XLS provides the module LEAID, which is installed as an abnormal termination exit (ATE) under LE. Abend-AID XLS gets control at the LEAID ATE to analyze and diagnose errors.

Language Environment provides its own extensive condition-handling. One function of that is to make sure registered ATEs are called by LE in the event of an error. LEAID must be registered as an LE ATE in order for Abend-AID XLS to be invoked when an error occurs and to create a diagnostic report. After Abend-AID XLS produces its report, LEAID returns control to LE.

Language Environment's TRAP run-time parameter specifies how LE handles abends and program interrupts. TRAP(ON) causes LE to trap abends and program interrupts. This prevents normal operating system abend and dump services from having an opportunity to process.

Notes:

1. Abend-AID XLS gets control after the CEE dump and so does not affect LE's response to a fault. You can specify LE responses ranging from no message to comprehensive dumps by setting a parameter in LE's Termination Thread Activity (TERMTHDACT) option.
2. Abend-AID XLS does not affect an application's response to a fault. You can specify a response of a return code or an abend with the parameters of LE's Abnormal Termination Enclave (ABTERMENC) option.

Invoking Abend-AID

Abend-AID can be invoked in Language Environment in two ways:

- **Implicit invocation:** Abend-AID is invoked implicitly at termination for unhandled conditions of severity 2 or greater. For these conditions, Language Environment error-handling calls all programs that have been registered as ATEs. If LEAID has been registered as an ATE, it is called by Language Environment and then invokes Abend-AID directly. Abend-AID is not invoked through SVC51 processing because in the LE environment itself, there is no abend to actually trigger MVS dump services.
- **Explicit invocation:** Abend-AID can be invoked explicitly by coding a call to LEAID within a user-written condition-handler. Refer to the *IBM Language Environment for MVS & VM Concepts Guide* for more information about Language Environment condition-handling.

Abend-AID cannot be invoked explicitly by coding a call to LEAID within an application program. A call to LEAID from an application program results in LEAID terminating immediately with a return code of 4. Instead, use the Abend-AID SNAP-AID facility to invoke Abend-AID explicitly from an application program outside of a user-written condition-handler. Refer to **Chapter 8, "Using SNAP-AID"** for more information.

Note: If LEAID is called from a user-written condition handler that does not handle the condition, two Abend-AID reports are produced. The first one results from the explicit call to LEAID coded within the condition handler. The second results from Language Environment calling LEAID as an ATE for the unhandled condition. Also, having multiple condition handlers and/or multiple ATEs can cause multiple Abend-AID reports to be produced for a single application error.

Routing Abend-AID Output

In Language Environment, the Abend-AID XLS report is written to an output destination chosen in the same manner as is done for normal Abend-AID XLS processing outside of Language Environment. The only exception to this is that the report cannot be written to the SYSUDUMP and SYSABEND IBM dump DDs. Normally, the Abend-AID XLS report is written using the site default report shared directory named by CWROUTE. However, this can be overridden as follows:

- //ABNLTERM DD, which overrides
- CWEXIT02, which overrides
- CWJOBTAB, which overrides
- a CWROUTE routing group specification, which overrides
- the CWROUTE site default report shared directory specification

If the //ABENDAID DD is not present in the application job step's JCL, Abend-AID's LE support dynamically allocates it. To write the Abend-AID XLS report to the //ABENDAID DD, use CWEXIT02, CWJOBTAB, or a CWROUTE routing group to designate that the report be written to SYSOUT. Refer to the *Abend-AID XLS Installation and Customization Guide*

LE Run-time Options

When an abnormal termination is encountered, the programmer can easily identify the LE run-time options in effect at the time of the failure, and see how and where each option was set. Abend-AID shows the setting for each option and the name of the specific option control table selected by LE. When a program is running, there are options that are obtained from many sources — the option control tables (CEEUOPT, CEEDOPT, CEEROPT) and from JCL overrides. Some options may overlap and/or conflict, making it difficult to determine why an application behaves as it does, including performance issues and how the job terminates if an error occurs.

LE User Heap Storage Analysis

Abend-AID displays the following for the LE User Heap Analysis:

1. User Heap run-time parameters and variables including initial heap size, increment size, number of segments, pointer to first segment, and pointer to last segment.
2. For each segment in the User Heap:
   a. Analysis summary for each segment in the User Heap:
      - number of elements
      - size of allocated storage
      - number of free elements
      - size of storage still available in the segment
      - number of element allocation errors

b. A display, by field, of each header in each segment.
c. A display of the allocation map for each segment. An entry is displayed for each element in each segment with its associated data address, data length, and status (allocated or free).
d. Validation for each element in a segment for allocation errors. If an allocation error is detected, the element is identified and the following is provided:
   - Storage surrounding the element in error
   - Storage that the elements in error are pointing to
   - Storage of the previous (last good) element, which may be suspect in overlay errors.

LE Run Time Library Services (RTLS)

LE run-time libraries contain modules that can be accessed via the system LNKLST, STEPLIB, and/or an LE services called RTLS (Run Time Library Services). Sites may choose a single method of access to the LE libraries or may allow a combination of these access mechanisms, depending on their needs. Typically, a combination method is needed when the site is converting to LE for the first time or upgrading LE and needs compatibility with older LE releases.

This varied access to these LE libraries may cause many run-time issues that may be the involved in and/or be the root cause of program faults. Abend-AID provides the type of library access and associated values that assist in debugging these types of problems. The LIBRARY(lib) and VERSION(ver) options that are used to select which RTLS logical library and version are displayed. See the OS/390 IBM manual, Language Environment for *z/OS Programming Reference* for more information on using RTLS.

Frequently Asked Questions

1. **Does Abend-AID work in Language Environment?**

   Yes! IBM, as originally requested by Compuware, provides the CEEEXTAN abnormal termination exit CSECT. Compuware uses the abnormal termination exit to allow Abend-AID to get control in order to produce an Abend-AID report. Abend-AID provides the module LEAID, which is installed as an abnormal termination exit (ATE). LEAID calls Abend-AID.

   An ATE is invoked by LE at program termination for all “unhandled conditions” of severity 2 or greater (return code 2000 or higher). LE can terminate with one of five severity codes, 0 through 4, which correspond to job step return codes of 0000 through 4000. For example, a data exception is a severity 3 condition. Prior to LE, a COBOL program that had a data exception would end with a S0C7 abend.

2. **What releases of LE does Abend-AID support?**

   Abend-AID XLS works with all releases of LE supported by IBM.

3. **How do I install Abend-AID in LE?**

   Installation requires ATE registration in LE. Refer to the *Abend-AID XLS Installation and Customization Guide* for details.

4. **How do I get my SYSUDUMP?**

   Your program usually does not abend in LE. Abend-AID provides an LE condition table that works in conjunction with the Abend-AID abend code tables. Using these tables, you can request that an IBM dump be written to the //SYSUDUMP DD. Another option is to add a //ABNLDUMP DD DUMMY to your JCL.

5. **How do I get my Abend-AID XLS reports written to my //SYSUDUMP DD?**

Unfortunately, you can't. Because no abend occurs, the //SYSUDUMP DD is not opened by MVS. Any product, other than MVS, opening the //SYSUDUMP DD can cause conflicts between MVS and that product. Abend-AID instead writes to an Abend-AID XLS report dataset and, if requested, the //ABENDAID DD, which will be dynamically allocated if it is not in the JCL.

6. **How do I specify the SYSOUT class when the Abend-AID report is written to the dynamically allocated //ABENDAID DD?**

   In the JES2 environment, dynamic allocation of the //ABENDAID DD is based on a model DD. The SYSOUT class and other subparameters of the model are used for the dynamically allocated //ABENDAID DD. If a model cannot be found, or if you are operating in a JES3 site, the //ABENDAID DD is dynamically allocated with a default SYSOUT class of *****. The default settings can be changed by updating global values using the Abend-AID XLS installation dialog. For information refer to "Dynamic Allocation Program (#XAAMKDD)" in the *Abend-AID XLS Installation and Customization Guide*.

7. **How do I get my job step to terminate so my JCL works the same as before LE?**

   LE provides a run-time option, ABTERMENC, which determines whether the job step ends with a return code or with an abend code. ABTERMENC affects only JCL processing and not dump processing.

   Abend-AID XLS ignores these abends.

8. **What is the TERMTHDACT run-time parameter UADUMP?**

   Starting with LE Release 5, the run-time parameter, TERMTHDACT(UADUMP), replaced the IBM-supplied abnormal termination exit, CEEBDATX. This parameter issues a U4039-0 abend that requests an IBM dump written to the SYSUDUMP DD. It is provided by IBM to debug LE failures only and to provide an IBM dump for systems programmers. It is not intended for use by application programmers to debug program problems.

9. **When I use CEEBDATX or the TERMTHDACT(UADUMP) run-time parameter, I always get an IBM dump regardless of the error. How can I get an IBM dump just for my S0C4 abends?**

   An Abend-AID LE condition table is available that controls whether an IBM dump is produced. When the table specifies that an IBM dump should be produced, LEAID issues a U4039-8 abend that requests an IBM dump written to the //SYSUDUMP DD. The LE condition table, CWTABL01, is used in conjunction with the Abend-AID abend code tables.

   Using these tables, you can request an IBM dump based upon the error condition and the use of CEEBDATX and TERMTHDACT (UADUMP) are not necessary. For example, you can request an IBM dump when the error condition is equivalent to a S0C4 abend. For more information refer to the *Abend-AID XLS Installation and Customization Guide*.

10. **If I already have an abnormal termination exit defined, can I still add LEAID to the table?**

    IBM allows you to define multiple abnormal termination exists. If there is more than one name defined, the abnormal termination exits are called in the order found in the CEEEXTAN CSECT.

Chapter 12.
Report Samples

This chapter shows samples of various report sections produced by Abend-AID XLS.

Accessing the Abend-AID XLS Report

The Abend-AID XLS Output Selection Menu, shown in Figure 12-1, lists all of the Abend-AID XLS report sections for online selection and viewing. It is part of the Compuware Viewing Facility (Compuware/VF).

Figure 12-1. Output Selection Menu

```
COMPUWARE/VF ---------------- OUTPUT SELECTION MENU ---------------------------
SELECT SECTION ===>

'AA.REPORT'        REPT.(L) 66  FLOCON01  J25586

   0 or HEADER   - System, job, and client information
   1 or DIAGS    - Diagnostic information
   2 or NSI      - Next Sequential Instruction (location in program)
   3 or TRACE    - Trace of program flow and program attributes
   5 or REGS     - Register contents and PSWs
   6 or PROG     - Program Storage
   7 or PLIST    - Program Listing (XLS only)
   8 or FILES    - Data management control blocks
   9 or IMS      - IMS diagnostic information and PCBs
  10 or IDMS     - CA-IDMS diagnostic and status information
  11 or DB2      - DB2 diagnostic and status information
  12 or SORT     - Current COBOL sort information
  13 or DATALOC  - COBOL Data Locator
  14 or MQ       - WebSphere MQ diagnostic information
  15 or LE       - LE information
  16 or EPILOG   - Abend-AID termination information

               Key section number or name and press ENTER.
    Or press ENTER to start at top of output and view sections 1 through 5.
```

Report Sections

Report sections are shown on the pages that follow.

Several sections have standard content regardless of the programming language: the Header, Registers, Trace, File, and Epilog sections. Other sections provide information specific to the programming language and database in use. Only those sections for which information is available at the time of the error are presented in the Abend-AID XLS report. For example, you will not have a DB2 section if the program does not access DB2.

Header Section

The Header section is the first section of the Abend-AID XLS report. This is a standard section that always provides the same type of information. The Header section identifies:

- Date and time of the error
- Job name
- Job number

- Step name
- Operating system release level
- Licensee name and number
- Abend-AID XLS release number.

Other information given in the Header section includes:

- CP FMID
- System on which your program was executing
- DFP release level
- JES2 release level
- CPU model number

Figure 12-2. Header section – COBOL

```
MONDAY     11 FEB 2002                  H09.04.00 F01/29/2002 R09.04.00
*************************************************************************
*              A b e n d - A I D   D i a g n o s t i c s                *
*************************************************************************
Copyright (C) 1976, 2002, Compuware Corporation.  Unpublished.
Rights Reserved Under The Copyright Laws Of The United States.

Licensed for use at: 000001
COMPUWARE CORPORATION

Online Technical Support available at:  frontline.compuware.com

JESID - JOB11861       OPSYS     - z/OS  1.1.0        Model   - 9672
Job   - AA94PLIC       DFSMS/MVS - V2R10M0           CP FMID - JBB7713
Step  - SAMPLEQ        JES2      - OS 2.10           System  - CW09
Time  - 14.52.39       LE        - OS/390 V2R10
```

Error Analysis Section

The Error Analysis section usually provides enough diagnostic information to resolve the problem. The cause of the error and corrective actions are described. The information varies, depending upon the programming language used. For external errors, like the S813-04 shown in Figure 12-3, the diagnosis includes:

- Cause of the error
- DDNAME and dataset name (whenever possible).

Figure 12-3. Error Analysis Section – S813 Example

```
An S813 abend occurred during execution of program PAYBKUP1.

*****************************************
*          Analysis of Error            *
*****************************************

The system completion code of 813-04 is invoked when the OPEN
routines of data management are unable to complete an OPEN macro
instruction for a magnetic tape data set.

Additional information can be found in the System Messages manual
under message IEC149I, return code 04.

Specific problem information for 813-04:

   Abending DDNAME is MASTER

DSNAME from DD card and HDR1 label on magnetic label
(VOLSER=TSOOOO  ) do not agree.
DSNAMES from DD card and HDR1 magnetic tape label -

DD CARD -                          TSOOO.PAYROLL.BKUP
TAPE    -                            TSOOO.PYRL.BKUP1

   ************************************************
   *   REMEMBER - Only the last 17 bytes of the   *
   *   HDR1 label and DD card DSN are compared.   *
   ************************************************
```

In the example above, the file name from the DD card did not match the name on the tape label. Abend-AID XLS provides a quick resolution to this problem. Change the JCL and resubmit the job.

Figure 12-4. Error Analysis Section – SFCC in Language Environment

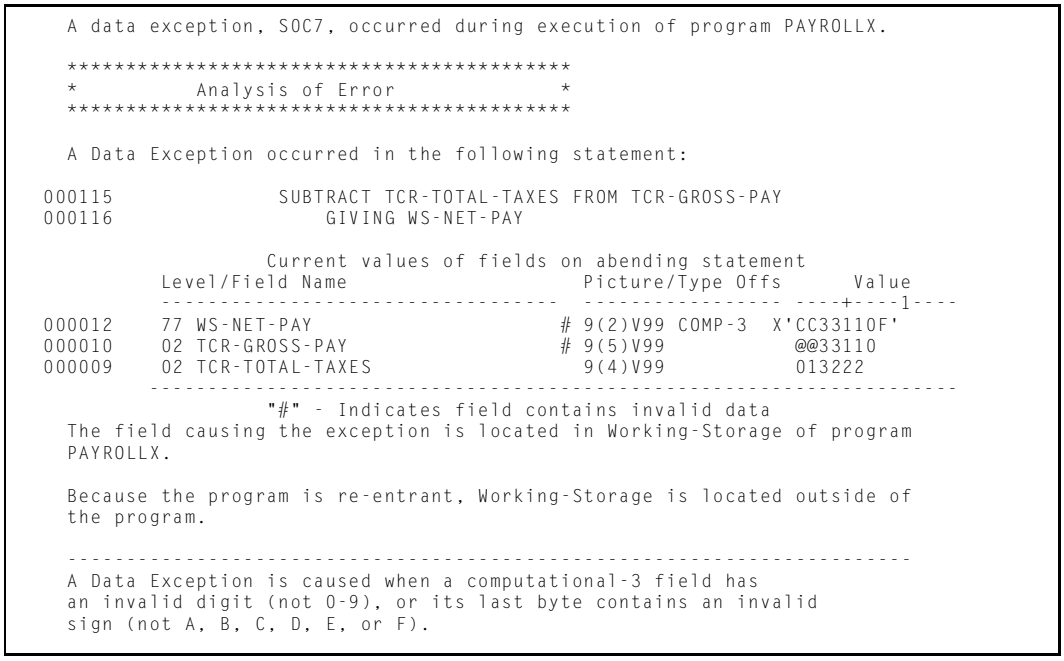
```
An SFCC abend occurred during execution of program REGMAIN1.

*****************************************
*          Analysis of Error            *
*****************************************

The LE release in use at execution time is higher than the LE release that
was used to link edit application.
```

In this example, an application program identified in the Header section was compiled with a higher level of LE than was used for execution. Abend-AID XLS identifies the type of fault, SFCA or SFCC, and the problem.

For data-related errors, the Error Analysis section provides:

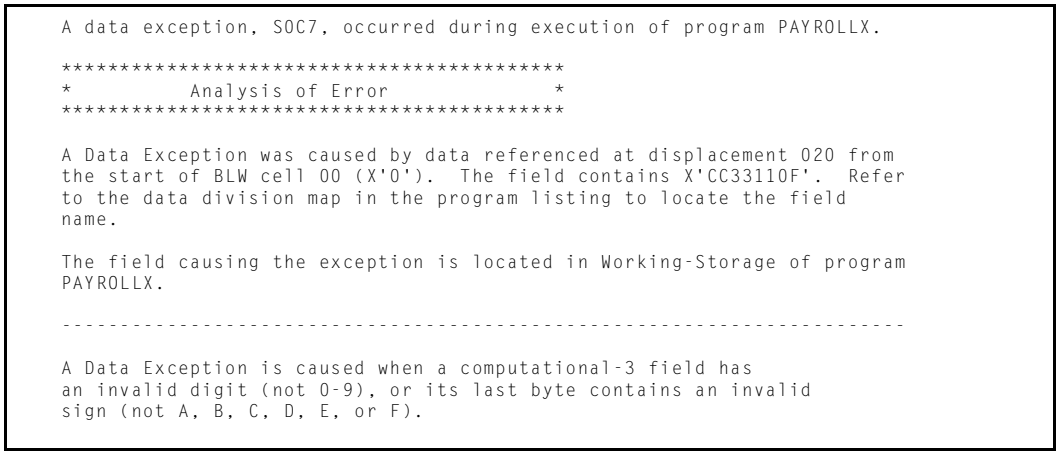
- Type of error
- Fields in error
- Location (displacement) of the fields within their respective base locator cell number
- Contents of the fields in error
- Description of the error.

Figure 12-5. Error Analysis Section with COBOL XLS



```
   A data exception, SOC7, occurred during execution of program PAYROLLX.

   ****************************************
   *          Analysis of Error           *
   ****************************************

   A Data Exception occurred in the following statement:

000115              SUBTRACT TCR-TOTAL-TAXES FROM TCR-GROSS-PAY
000116                  GIVING WS-NET-PAY

                  Current values of fields on abending statement
          Level/Field Name                  Picture/Type Offs      Value
          --------------------------------  ----------------- ----+----1----
000012    77 WS-NET-PAY                    # 9(2)V99 COMP-3  X'CC33110F'
000010    02 TCR-GROSS-PAY                 # 9(5)V99           @@3311O
000009    02 TCR-TOTAL-TAXES                 9(4)V99           013222
          ------------------------------------------------------------------------
                    "#" - Indicates field contains invalid data
   The field causing the exception is located in Working-Storage of program
   PAYROLLX.

   Because the program is re-entrant, Working-Storage is located outside of
   the program.

   ------------------------------------------------------------------------
   A Data Exception is caused when a computational-3 field has
   an invalid digit (not 0-9), or its last byte contains an invalid
   sign (not A, B, C, D, E, or F).
```

Figure 12-6. Error Analysis Section with COBOL Basic Support



```
   A data exception, SOC7, occurred during execution of program PAYROLLX.

   ****************************************
   *          Analysis of Error           *
   ****************************************

   A Data Exception was caused by data referenced at displacement 020 from
   the start of BLW cell 00 (X'0').  The field contains X'CC33110F'.  Refer
   to the data division map in the program listing to locate the field
   name.

   The field causing the exception is located in Working-Storage of program
   PAYROLLX.

   ------------------------------------------------------------------------

   A Data Exception is caused when a computational-3 field has
   an invalid digit (not 0-9), or its last byte contains an invalid
   sign (not A, B, C, D, E, or F).
```

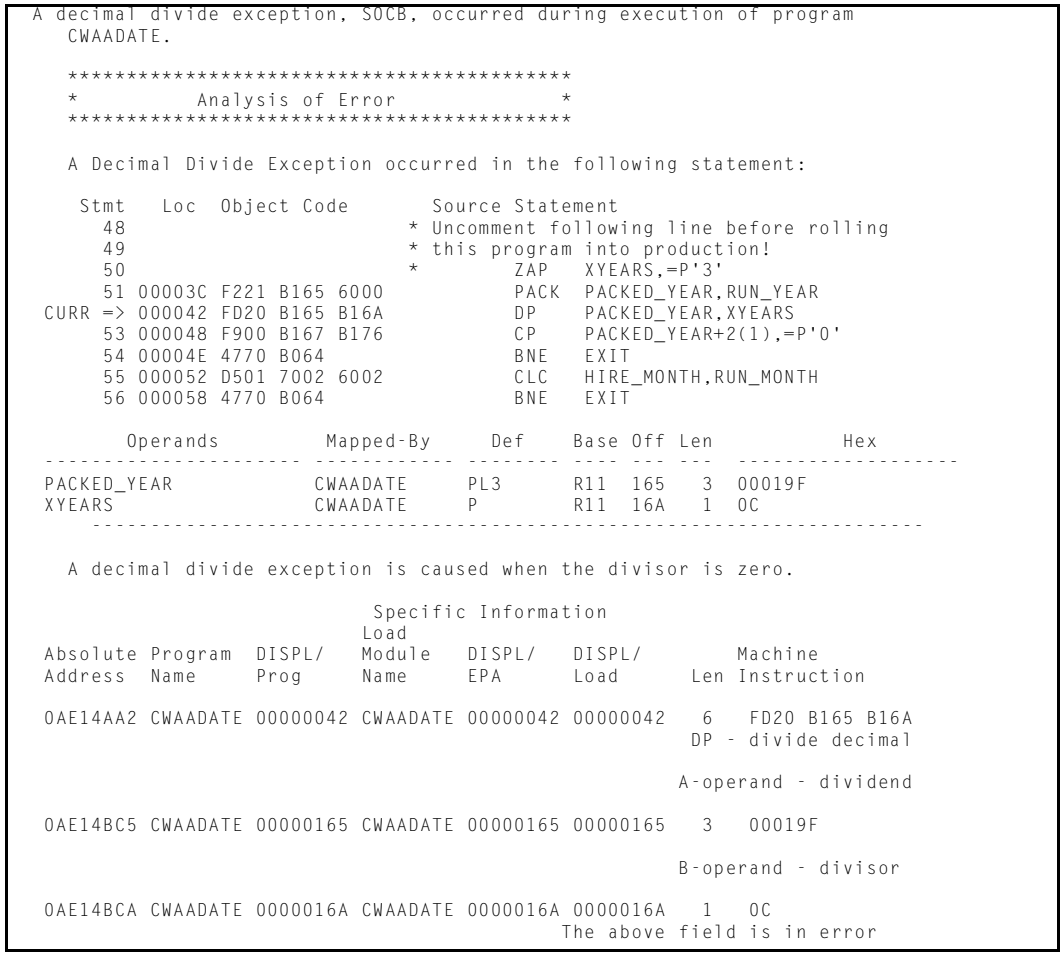
Figure 12-7. Error Analysis Section with PL/I XLS

```
   The user abend code of U3001 was issued by PL/I after normal return from
   an error or finish on-unit for error on-code 0320.  That on-code
   corresponds to the OS completion code of S0CB.

   ****************************************
   *          Analysis of Error           *
   ****************************************

   A Decimal Divide Exception occurred in the following statement:

000065             AVG_HOURS = TOTAL_HOURS / MONTH_CODE;

               Current values of fields on abending statement
--- Variable/Attributes -----------    Value----1----+----2----+----3----+----4

CALC_PAYROLL:AVG_HOURS
    AUTO    FIXED DEC(5,1)             +43.0
CALC_HOURS:TOTAL_HOURS
    STATIC  FIXED DEC(5)               +101
PAYROLL:TIME_RECORD.MONTH_CODE
    AUTO    PIC'99'                    00
--------------------------------------------------------------------------------
                "#" - Indicates field contains invalid data
```

Figure 12-8. Error Analysis Section with PL/I Basic Support

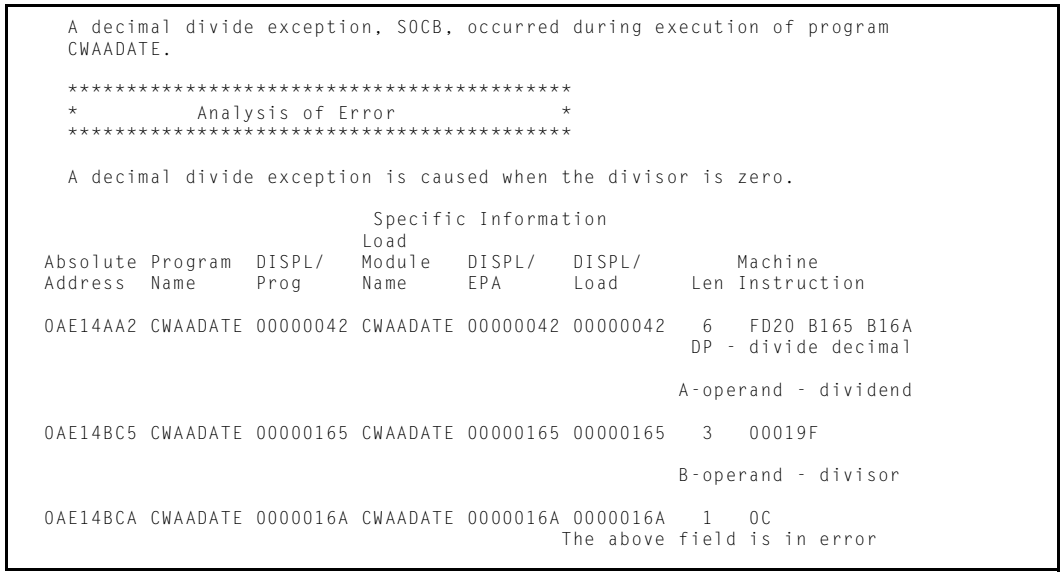
```
   The user abend code of U3001 was issued by PL/I after normal return from
   an error or finish on-unit for error on-code 0320.  That on-code
   corresponds to the OS completion code of S0CB.

   ****************************************
   *          Analysis of Error           *
   ****************************************

   A decimal divide exception is caused when the divisor is zero.

                          Specific Information
         CSECT    DISPL/   Module   DISPL/   DISPL/
  ADDR   Name     CSECT    Name      EPA      Load    Len Instruction
06900F2E PAYROLL1 000009DE PAYEXEC  00000A5E 00000A5E  6  FD91 D098 D0C8
                                                        DP - divide decimal

                                                     A-operand - dividend

00011F18                                              10  0000001010000000000000C

                                                     B-operand - divisor

00011F48                                                2  000C
                                                  The above field is in error

Note: The CSECT and module displacements are not applicable
      because the operands are located in storage outside
      of the program.
```

Figure 12-9. Error Analysis Section with Assembler XLS



```
A decimal divide exception, SOCB, occurred during execution of program
   CWAADATE.

   ******************************************
   *          Analysis of Error             *
   ******************************************

   A Decimal Divide Exception occurred in the following statement:

    Stmt   Loc  Object Code       Source Statement
      48                       * Uncomment following line before rolling
      49                       * this program into production!
      50                       *        ZAP   XYEARS,=P'3'
      51 00003C F221 B165 6000          PACK  PACKED_YEAR,RUN_YEAR
CURR => 000042 FD20 B165 B16A           DP    PACKED_YEAR,XYEARS
      53 000048 F900 B167 B176          CP    PACKED_YEAR+2(1),=P'0'
      54 00004E 4770 B064               BNE   EXIT
      55 000052 D501 7002 6002          CLC   HIRE_MONTH,RUN_MONTH
      56 000058 4770 B064               BNE   EXIT

        Operands         Mapped-By     Def    Base Off Len           Hex
----------------------- ------------ -------- ---- --- ---  --------------------
PACKED_YEAR            CWAADATE     PL3      R11  165   3  00019F
XYEARS                 CWAADATE     P        R11  16A   1  0C
    ------------------------------------------------------------------------

   A decimal divide exception is caused when the divisor is zero.

                            Specific Information
                           Load
Absolute Program  DISPL/   Module   DISPL/   DISPL/        Machine
Address  Name     Prog     Name     EPA      Load      Len Instruction

0AE14AA2 CWAADATE 00000042 CWAADATE 00000042 00000042   6   FD20 B165 B16A
                                                          DP - divide decimal

                                                        A-operand - dividend

0AE14BC5 CWAADATE 00000165 CWAADATE 00000165 00000165   3   00019F

                                                        B-operand - divisor

0AE14BCA CWAADATE 0000016A CWAADATE 0000016A 0000016A   1   0C
                                                   The above field is in error
```

Figure 12-10. Error Analysis Section with Assembler Basic Support



```
   A decimal divide exception, SOCB, occurred during execution of program
   CWAADATE.

   ******************************************
   *          Analysis of Error             *
   ******************************************

   A decimal divide exception is caused when the divisor is zero.

                            Specific Information
                           Load
Absolute Program  DISPL/   Module   DISPL/   DISPL/        Machine
Address  Name     Prog     Name     EPA      Load      Len Instruction

0AE14AA2 CWAADATE 00000042 CWAADATE 00000042 00000042   6   FD20 B165 B16A
                                                          DP - divide decimal

                                                        A-operand - dividend

0AE14BC5 CWAADATE 00000165 CWAADATE 00000165 00000165   3   00019F

                                                        B-operand - divisor

0AE14BCA CWAADATE 0000016A CWAADATE 0000016A 0000016A   1   0C
                                                   The above field is in error
```

Error Location Section

The Error Location section provides information that can be used to locate the statement in error. Information presented in this section includes:

- Program's compile date
- Program's link-edit date
- Program name and module lengths
- Load module name and the load library name
- Location of the last I/O operation or subroutine call, if applicable.

Figure 12-11. Error Location Section with COBOL XLS

```
   ******************************************
   *             Error Location             *
   ******************************************

   The statement being executed was:

 000115              SUBTRACT TCR-TOTAL-TAXES FROM TCR-GROSS-PAY
 000116                  GIVING WS-NET-PAY

   This statement is contained in
   paragraph "READ-A-RECORD" in program "PAYROLLX".

   The program was compiled on 08 FEB 2002 and is 00000FB8 bytes long.

   It is part of load module PAYLOAD.

   The module was loaded from STEPLIB library CW.AA.TEMPLOA7.

   The module was link edited on 08 FEB 2002 and is 00001400 bytes long.

   The last known I/O request or call in the program above was:

 000101            READ TIMECARD INTO WS-TIMECARD-REC.

   This statement is contained in
   paragraph "READ-A-RECORD" in program "PAYROLLX".
```

Figure 12-12. Error Location Section with COBOL Basic Support

```
   ******************************************
   *             Error Location             *
   ******************************************

   The next sequential instruction to be executed in program PAYROLLX was
   at displacement 0000065A.

   The program was compiled on 08 FEB 2002 and is 00000FB8 bytes long.

   It is part of load module PAYLOAD.

   The module was loaded from STEPLIB library CW.AA.TEMPLOA7.

   The module was link edited on 08 FEB 2002 and is 00001400 bytes long.

   The last known I/O request or call in the program above was issued with
   a return address at displacement 000005AC.
```

Figure 12-13. Error Location Section with PL/I XLS

```
   ****************************************
   *             Error Location           *
   ****************************************

   The statement being executed was:

000065             AVG_HOURS = TOTAL_HOURS / MONTH_CODE;

   This statement is contained in
   procedure PAYROLL:CALC_PAYROLL:CALC_HOURS.

   The program was compiled on 08 FEB 2002 and is 00000A28 bytes long.

   It is part of load module PAYEXEC.

   The module was loaded from STEPLIB library CW.AA.TEMPLOA7.

   The module was link edited on 08 FEB 2002 and is 00003B30 bytes long.

   The last known I/O request or call in the program above was:

000063              OVERTIME_TOTAL.ACCRUED_HOURS = OVERTIME_TOTAL.ACCRUED_HOURS +
                        OVERTIME_HOURS_WEEKLY;

   This statement is contained in
   procedure PAYROLL:CALC_PAYROLL:CALC_HOURS.
```

Figure 12-14. Error Location Section with PL/I Basic Support

```
   ****************************************
   *             Error Location           *
   ****************************************

   The next sequential instruction to be executed was in statement 65 at
   offset 00000016C in procedure CALC_HOURS.

   The program was compiled on 08 FEB 2002 and is 00000A28 bytes long.

   It is part of load module PAYEXEC.

   The module was loaded from STEPLIB library CW.AA.TEMPLOA7.

   The module was link edited on 08 FEB 2002 and is 00003B30 bytes long.

   The last known I/O request or call in the program above was issued with
   a return address at displacement 0000097C.
```

Figure 12-15. Error Location Section with Assembler XLS

```
   ****************************************
   *             Error Location           *
   ****************************************

   The statement being executed in CSECT CWAADATE was at displacement
   00000042.

    Stmt   Loc  Object Code       Source Statement
      48                        * Uncomment following line before rolling
      49                        * this program into production!
      50                        *        ZAP   XYEARS,=P'3'
      51 00003C F221 B165 6000           PACK  PACKED_YEAR,RUN_YEAR
CURR => 000042 FD20 B165 B16A            DP    PACKED_YEAR,XYEARS
      53 000048 F900 B167 B176           CP    PACKED_YEAR+2(1),=P'0'
      54 00004E 4770 B064                BNE   EXIT
      55 000052 D501 7002 6002           CLC   HIRE_MONTH,RUN_MONTH
      56 000058 4770 B064                BNE   EXIT

   The program was compiled on 08 FEB 2002 and is 00000178 bytes long.

   It is part of load module CWAADATE.

   The module was loaded from STEPLIB library CW.AA.TEMPLOA7.

   The module was link edited on 08 FEB 2002 and is 00000178 bytes long.
```

Figure 12-16. Error Location Section with Assembler Basic Support

```
   ****************************************
   *             Error Location           *
   ****************************************

   The next sequential instruction to be executed in program CWAADATE
   was at displacement 00000048.

           00000038  92E8B164      MVI   356(R11),X'E8'
           0000003C  F221B1656000  PACK  357(3,R11),0(2,R6)
           00000042  FD20B165B16A  DP    357(3,R11),362(1,R11)
    NSI==> 00000048  F900B167B176  CP    359(1,R11),374(1,R11)
           0000004E  4770B064      BNE   100(,R11)
           00000052  D50170026002  CLC   2(2,R7),2(R6)
           00000058  4770B064      BNE   100(,R11)

   The program was compiled on 08 FEB 2002 and is 00000178 bytes long.

   It is part of load module CWAADATE.

   The module was loaded from STEPLIB library CW.AA.TEMPLOA7.

   The module was link edited on 08 FEB 2002 and is 00000178 bytes long.
```

Trace Section

The Trace section is a standard section that shows you which programs were called and in what order. Also provided in this section are the Application Program Attributes for all application programs in the Call Trace Summary. The Trace section gives the following information:

- Called/linked programs on the save area chain
- Program location(s) where the call(s) occurred
- Program in error, when available
- Library that each load module in the Call Trace was loaded from
- Application Program Attributes listing:
  - Each program's name and its load module
  - Compile date, length, and language for each program.

Figure 12-17 on page 12-10 shows a sample Trace section.

Figure 12-17. Trace Section

```
*****************************************
*           Call Trace Summary          *
*****************************************

 ******Calling*****     ****Return****           ******Called******
 Load-Mod  Program      Type     Value           Program   Load-Mod

 *SYSTEM                                 Links to CWABTST   CWABTST    *

 CWABTEST   CWABTEST    DISP  000001AA  Calls              IGG019DJ

                                                 Program Causing Error  *

 ---------------------------------------------------------------------
 Load-Mod    Amode  Rmode  Resides in
 CWABTEST     24     24    AA.R903.MXA.LOAD
 IGG019DJ     24     24    PLPA
 ---------------------------------------------------------------------


 *****************************************
 *     Application Program Attributes      *
 *****************************************

 Load-Mod   Program     Compile Date  Length     Language

 CWABTEST   CWABTEST    04 APR 2000   00000AAD   HLASM
```

Registers Section

The Registers section is a standard section that displays Supporting Environmental Data, which identifies:

- Abending program status word (PSW) and program PSW
- Entry point (EPA) and load point addresses (LPA)
- Instruction length code (ILC)
- Register contents and descriptions at the time of the error
- Load module name.

Figure 12-18. Registers Section

```
   ****************************************
   *      Supporting Environmental Data      *
   ****************************************

   PARM Data - HEAP(400,400,ANY),ABTERMENC(RETCODE),RPTSTG(ON)/
   Length of data -
  48

 Abend PSW - 078D1000 8000690A   A(PAYROLLX) + 00000542
 Prog PSW  - 078D1000 8000690A   A(PAYROLLX) + 00000542
 Load Module - PAYROLLX                  Entry Point Address - 00054958
 ILC - 06,   INTC - 07                   Load  Point Address - 00054958

  Registers at time of error  (Descriptions based on 31 bit addresses)

REG HEX       Decimal         Description
R0  00000001                1
R1  09BF5030      163,532,848  A(09BF4000) + 00001030, PVT SP=000 ALLOC
R2  000177EC           96,236  A(00017000) + 000007EC, PVT SP=001 ALLOC
R3  00006D00           27,904  A(PAYROLLX) + 00000938
R4  00006FC0           28,608  A(PAYROLLX) + 00000BF8
R5  80006860  -2,147,456,928  A(PAYROLLX) + 00000498
R6  09BF8270      163,545,712  A(09BF8000) + 00000270, PVT SP=000 ALLOC
R7  09BF5030      163,532,848  A(09BF4000) + 00001030, PVT SP=000 ALLOC
R8  00007161           29,025  A(PAYROLLX) + 00000D99
R9  00007220           29,216  A(PAYROLLX) + 00000E58
R10 000064C4           25,796  A(PAYROLLX) + 000000FC
R11 80010330  -2,147,417,296  A(CEEBBEXT) + 00000000
R12 00017B10           97,040  A(00017000) + 00000B10, PVT SP=001 ALLOC
R13 00006B28           27,432  A(PAYROLLX) + 00000760
R14 8004C3DC  -2,147,171,364  A(IGZEQBL ) + 00000604
R15 00DD01E0       14,483,936  A(IGG019AA) + 00000000
```

Program Storage Section

The Program Storage section formats program storage for application programs on the calling chain.

Figure 12-19. Program Storage Section with COBOL XLS

```
             Working-Storage Section
Level/Field Name                    Picture/Type Offs      Value
----------------------------------  ----------------- ----+----1----+----2
77 WS-LINE-CTR                      S999     COMP-3   +4
77 WS-PAGE-CTR                      S999     COMP-3   +1
77 WS-RECORD-CTR                    S999     COMP-3   +4
77 WS-TOTAL-NET-PAY                 S9(7)V99 COMP-3   +339117
77 WS-NET-PAY                     # S9(5)V99 COMP-3 X'CC33110F'
77 WS-EMP-NO                        9(8)             04060293
01 WS-TIMECARD-REC
 02 TCR-EMP-NO                      9(8)             04060293
 02 TCR-MONTH-CODE                  99               02
 02 TCR-EMP-NAME                    X(32)            EDWARDS, ALEXANDER
 02 TCR-GROSS-PAY                 # 9(5)V99         X'7C7CF3F3F1F1F0'
 02 TCR-TOTAL-TAXES                 9(4)V99          013222

             Linkage Section
Level/Field Name                    Picture/Type Offs      Value
----------------------------------  ----------------- ----+----1----+----2
01 PARMDATA
 02 PARM-LEN                        9(3)     COMP     7
 02 PARM-STRING                     X(7)             SPACES
-------------------------------------------------------------------------------

                              Supporting data
                           Contents of BLW cells
    BLW      0 (X'0')    - 00007220

                           Contents of BLL cells
    BLL      0 (X'0')    - 00000000   BLL      1 (X'1')    - 00013808

                           Contents of BLF cells
    BLF      0 (X'0')    - 09BF8270   BLF      1 (X'1')    - 00007161
    BLF      2 (X'2')    - 09BF5030

                           Task Global Table (TGT)
 DSPL Address   --------------- Data ---------------
00000 00006B28  00108001 00015E98 00000000 80006A52  ......;q......].
00010 00006B38  89BD52D8 8004C3DC 00DD01E0 00000001  i..Q..C....\....
00020 00006B48  00006F60 000177EC 00006D00 00006FC0  ..?-......_...?{
00030 00006B58  80006860 09BF8270 09BF4FE0 00007161  ...-..b...|\.../
00040 00006B68  0000680C 000064BC F3E3C7E3 00026018  ........3TGT..-.
```

Figure 12-20. Program Storage Section with COBOL Basic Support

```
              Working-Storage Section

Working storage referenced by BLW cell      0 (X'0')


 DSPL Address   -------------- Data ---------------
00000 00007220  004C0000 00000000 001C0000 00000000  .<..............
00010 00007230  004C0000 00000000 00033911 7C000000  .<..........@...
00020 00007240  CC3311OF 00000000 F0F4F0F6 F0F2F9F3  ........04060293
00030 00007250  F0F4F0F6 F0F2F9F3 F0F2C5C4 E6C1D9C4  0406029302EDWARD
00040 00007260  E26B40C1 D3C5E7C1 D5C4C5D9 40C74B40  S, ALEXANDER G.
00050 00007270  40404040 40404040 40407C7C F3F3F1F1           @@3311
00060 00007280  F0F0F1F3 F2F2F2F4 F3F2F3F4 F3F2F440  001322243234324
00070 00007290  40404040 40404040 40404040 40404040

                               Linkage Section

Linkage storage referenced by BLL cell      1 (X'1')
 (Length:  DEFAULT - 4K)

 DSPL Address   -------------- Data ---------------
00000 00013808  00074040 40404040 4061E2E3 C1C50000  ..       /STAE..
00010 00013818  00000000 00000000 00000000 00000000  ................

--------------------------------------------------------------------------------

                                Supporting data

                            Contents of BLW cells

    BLW      0 (X'0')    - 00007220

                            Contents of BLL cells

    BLL      0 (X'0')    - 00000000   BLL      1 (X'1')    - 00013808

                            Contents of BLF cells

    BLF      0 (X'0')    - 09BF8270   BLF      1 (X'1')    - 00007161
    BLF      2 (X'2')    - 09BF5030

                            Task Global Table (TGT)

 DSPL Address   -------------- Data ---------------
00000 00006B28  00108001 00015E98 00000000 80006A52  ......;q......].
00010 00006B38  89BD52D8 8004C3DC 00DD01E0 00000001  i..Q..C....\....
00020 00006B48  00006F60 000177EC 00006D00 00006FC0  ..?-......_...?{
00030 00006B58  80006860 09BF8270 09BF4FE0 00007161  ...-..b...|\.../
00040 00006B68  0000680C 000064BC F3E3C7E3 00026018  ........3TGT..-.
```

Figure 12-21. Program Storage Section with PL/I XLS

```
                  Variable Storage - Procedure CALC_PAYROLL
--- Variable/Attributes -----------    Value----1----+----2----+----3----+----4

AVG_HOURS
    AUTO    FIXED DEC(5,1)             +43.0

1   CNTL_STRUCT                (0)
    CTL
 2  TABLE_NAME
            CHAR(30)                   JOHNSTONE, REGINALD B.
 2  TABLE_DATA (1:2)
            (1)  FLT DEC(5)            +2.000000E+0
            (2)  FLT DEC(5)            +2.000000E+0
 2  TABLE_ACTIVITY
            CHAR(18)                   02RECORD 2

1   CNTL_STRUCT               (-1)
    CTL
 2  TABLE_NAME
            CHAR(30)                   JONES,  KERI, M.
 2  TABLE_DATA (1:2)
            (1)  FLT DEC(5)            +1.000000E+0
            (2)  FLT DEC(5)            +1.000000E+0
 2  TABLE_ACTIVITY
            CHAR(18)                   01RECORD 1

1   INVENTORY_RANGE (1:3)
    AUTO
 2  (1)  WAREHOUSE
 3  MAX_STOCK
            (1)  FIXED DEC(4,1)        +99.0
 3  MIN_STOCK
            (1)  FIXED DEC(3,1)        +88.0
```

Note: PL/I basic support and XLS do not support based variables with the ALLOCATE statement.

Figure 12-22. Program Storage Section with PL/I Basic Support

```
            ******************************************************
            Static storage for PL/I procedure PAYROLLX
            in CSECT PAYRLLX1 of load module PAYEXEC
            Compiled with options:  SYSTEM( MVS )
            ******************************************************

 DSPL Address   -------------- Data ---------------
00000 06901800  E0000DC4 06900C64 06900D32 06900FC4  \..D.........D
00010 06901810  069016D8 0690173A 06901750 069017AA  ...Q.......&....
00020 06901820  069017AA 069017AA 069017AA 069017AA  ................


00410 06901C10  40404040 E3D6E3C1 D340D5C5 E340D7C1      TOTAL NET PA
00420 06901C20  E8D5C5E3 40D7C1E8 40404040 40404040  YNET PAY
00430 06901C30  E3D6E3C1 D340E3C1 E7C5E240 40404040  TOTAL TAXES
00440 06901C40  40C7D9D6 E2E240D7 C1E84040 40404040   GROSS PAY
00450 06901C50  D4D6D5E3 C8404040 40404040 40404040  MONTH
00460 06901C60  40C5D4D7 D3D6E8C5 C540D5E4 D4C2C5D9   EMPLOYEE NUMBER


            ******************************************************
            Automatic storage for PL/I procedure PAYROLLX
            which issued CALL or BEGIN at offset
            relative to main entry of procedure
            That offset was located at statement 21003FA
            ******************************************************

 DSPL Address   -------------- Data ---------------
00000 00011888  E0250000 000114D0 00000000 8690105E  \......}....f..;
00010 00011898  06903B08 00011DF8 06901BAC 86900FC4  .......8....f..D
00020 000118A8  06901800 069025F8 00000000 00011D92  .......8.......k


002F0 00011B78  F4F3F2F4 40404040 40404040 40404040  4324
00300 00011B88  40404040 4040C5C4 E6C1D9C4 E26B40C1        EDWARDS, A
00310 00011B98  D3C5E7C1 D5C4C5D9 40404040 40404040  LEXANDER
00320 00011BA8  40404040 40404040 40404040 F0F4F0F6              0406
00330 00011BB8  F0F2F9F3 40404040 40404040 40404040  0293
00340 00011BC8  40404040 40404040 40404040 5BF1F26B              $12,
00350 00011BD8  F0F3F34B F3F24040 4040405B F36BF1F2  033.32     $3,12
00360 00011BE8  F94BF9F2 40404040 40404040 5B40F86B  9.92         $ 8,
00370 00011BF8  F9F0F34B F4F04040 40404040 40404040  903.40


                ************************************
                * Program storage for CSECT PLISTART *
                * located in load module PAYEXEC     *
                ************************************

 DSPL Address   -------------- Data ---------------
00000 06900BD8  47000000 47000002 90ECD00C 053047F0  .........}....0
00010 06900BE8  30180014 CE0A0203 06900C04 D7D3C9E2  .............PLIS
00020 06900BF8  E3C1D9E3 000058F0 F070050F 069025D0  TART...00......}
00030 06900C08  069025D8 00000000 00000000 0000000C  ...Q............
00040 06900C18  FFFF0044 00000000 00000000 00000000  ................
```

Note: PL/I basic support and XLS do not support based variables with the ALLOCATE statement.

Figure 12-23. Program Storage Section with Assembler XLS

```
 STORAGE FOR CSECT CWAADATE (R11)
Offset Address      Line  Label    Type        ----- HEX ------ - CHAR -
------ -------- --------- -------- --------    ---------------- --------
000000 0AE14A60       6             DS 0H
0000EC 0AE14B4C     125    SAVEAREA DS 18F      0000000000030018 ........
0000F4 0AE14B54                                 0000000000000000 ........
                    132    DATE_DAYS
000134 0AE14B94                      DS 0C
000134 0AE14B94     133              DC C        F0F1F3F1         0131
000138 0AE14B98     134              DC C        F0F2F2F8         0228
00013C 0AE14B9C     135              DC C        F0F3F3F1         0331
000140 0AE14BA0     136              DC C        F0F4F3F0         0430
000144 0AE14BA4     137              DC C        F0F5F3F1         0531
000148 0AE14BA8     138              DC C        F0F6F3F0         0630
00014C 0AE14BAC     139              DC C        F0F7F3F1         0731
000150 0AE14BB0     140              DC C        F0F8F3F1         0831
000154 0AE14BB4     141              DC C        F0F9F3F0         0930
000158 0AE14BB8     142              DC C        F1F0F3F1         1031
00015C 0AE14BBC     143              DC C        F1F1F3F0         1130
000160 0AE14BC0     144              DC C        F1F2F3F1         1231
                    145    CHECKED_FOR_EOM_SW
000164 0AE14BC4                      DC C        E8               Y
                    146    PACKED_YEAR
000165 0AE14BC5                      DS PL3      00019F           ...
                    147    PACKED_YEAR2
000168 0AE14BC8                      DS PL2      0000             ..
00016A 0AE14BCA     148    XYEARS    DC P        0C               .
                                     DS CL1
                    149    CALC_EOM_RETURN@
00016C 0AE14BCC                      DS F        8AE14A98         ..6q
                         LTORG
000170 0AE14BD0     151                =C        F0F2             02
000172 0AE14BD2     152                =C        F2F8             28
000174 0AE14BD4     153                =C        F2F9             29
000176 0AE14BD6     154                =P        0C               .
000177 0AE14BD7     155                =P        4C               <

 STORAGE FOR DSECT #EOM_SW (R4)
Offset Address      Line  Label    Type        ----- HEX ------ - CHAR -
------ -------- --------- -------- --------    ---------------- --------
000000 0000F045     157    EOM_SW   DS C        D5               N

 STORAGE FOR DSECT #ANNIV_IND (R5)
Offset Address      Line  Label    Type        ----- HEX ------ - CHAR -
------ -------- --------- -------- --------    ---------------- --------
                    159    ANNIV_IND
000000 0000F046                      DS C        D5               N

 STORAGE FOR DSECT #RUN_DATE (R6)
Offset Address      Line  Label    Type        ----- HEX ------ - CHAR -
------ -------- --------- -------- --------    ---------------- --------
000000 0000F070     161    RUN_DATE DS 0CL6
000000 0000F070     162    RUN_YEAR DS CL2      F1F9             19
                    163    RUN_MONTH
000002 0000F072                      DS CL2      F9F8             98
000004 0000F074     164    RUN_DAY  DS CL2      F0F4             04
```

Figure 12-24. Program Storage Section with Assembler Basic Support

```
                    ************************************
                    * Program storage for CSECT CWAADATE *
                    * located in load module CWAADATE    *
                    * Length is 00000178                 *
                    * Language is HLASM                  *
                    ************************************

*********************************************************************
*               Save area used to call CSECT CWAADATE               *
*********************************************************************
* Located at:  00030018  A(0002C000) + 00004018, PVT SP=001 ALLOC  *
* Descriptions based on 31 bit addresses                            *
*********************************************************************
    HEX       Decimal         Description
WD1 00102401       1,057,793  UNALLOCATED - LSQA
BCK 0000E858          59,480  A(CWAACOB ) + 000087B8
FWD 00030320         197,408  A(0002C000) + 00004320, PVT SP=001 ALLOC
R14 800244A8  -2,147,335,000  A(IGZCLNK ) + 00000410
R15 8AE14A60  -1,964,946,848  A(CWAADATE) + 00000000
R0  00030320         197,408  A(0002C000) + 00004320, PVT SP=001 ALLOC
R1  0000EAD0          60,112  A(CWAACOB ) + 00008A30
R2  00021719         136,985  A(00021000) + 00000719, PVT SP=001 ALLOC
R3  0000EAD0          60,112  A(CWAACOB ) + 00008A30
R4  0000654B          25,931  A(CWAACOB ) + 000004AB
R5  00000004               4
R6  000507B8         329,656  A(0002C000) + 000247B8, PVT SP=001 ALLOC
R7  0000E9B0          59,824  A(CWAACOB ) + 00008910
R8  800244AA  -2,147,334,998  A(IGZCLNK ) + 00000412
R9  0000E858          59,480  A(CWAACOB ) + 000087B8
R10 00050038         327,736  A(0002C000) + 00024038, PVT SP=001 ALLOC
R11 80024098  -2,147,336,040  A(IGZCLNK ) + 00000000
R12 0001F6A0         128,672  A(0001D000) + 000026A0, PVT SP=001 ALLOC

*********************************************************************
*                      Storage of CSECT CWAADATE                    *
*********************************************************************
 DSPL Address   -------------- Data ---------------
00000 0AE14A60  90ECD00C 18BF41B0 B00050D0 B0F041D0  ..}.......&}.0.}
00010 0AE14A70  B0EC1831 58403000 58503004 58603008  ..... ...&...-..
00020 0AE14A80  5870300C 41707000 92D55000 95D5B164  ........kN&.nN..
00030 0AE14A90  4770B03C 4DA0B06E 92E8B164 F221B165  ....(..>kY..2...
00040 0AE14AA0  6000FD20 B165B16A F900B167 B1764770  -.......-9.......
00050 0AE14AB0  B064D501 70026002 4770B064 92E85000  ..N...-.....kY&.
00060 0AE14AC0  47F0B064 58D0B0F0 98ECD00C 07FE50A0  .0...}.0q.}...&.
00070 0AE14AD0  B16CD501 6002B170 4780B0AC 4180000C  .%N.-...........
00080 0AE14AE0  4190B134 D5019000 60024780 B09A4190  ....N...-.......
00090 0AE14AF0  90044680 B08447F0 B0ACD501 90026004  .....d.0..N...-.
000A0 0AE14B00  4770B0A8 92E84000 47F0B0E6 F221B165  ...ykY ..0.W2...
000B0 0AE14B10  6000FD20 B165B177 F900B167 B1764780  -.......9.......
000C0 0AE14B20  B0D4D501 6004B172 4770B0E6 92E84000  .MN.-......WkY .
000D0 0AE14B30  47F0B0E6 D5016004 B1744770 B0E692E8  .0.WN.-......WkY
000E0 0AE14B40  400047F0 B0E658A0 B16C07FA 00000000   ..0.W...%......
000F0 0AE14B50  00030018 00000000 00000000 00000000  ................
00100 0AE14B60  00000000 00000000 00000000 00000000  ................
            LINES 0AE14B70-0AE14B80 SAME AS ABOVE
00130 0AE14B90  00000000 F0F1F3F1 F0F2F2F8 F0F3F3F1  ....013102280331
00140 0AE14BA0  F0F4F3F0 F0F5F3F1 F0F6F3F0 F0F7F3F1  0430053106300731
00150 0AE14BB0  F0F8F3F1 F0F9F3F0 F1F0F3F1 F1F1F3F0  0831093010311130
00160 0AE14BC0  F1F2F3F1 E800019F 00000C00 8AE14A98  1231Y.........Ûq
00170 0AE14BD0  F0F2F2F8 F2F90C4C                    022829.<
```

Program Listing Section

Available only with XLS. Displays the program source code and identifies the *current statement*. The current statement in the program in error is either the actual statement in error or the last call. The current statement in any program other than the program in error indicates the last known call in that program.

Figure 12-25. Program Listing Section with COBOL XLS

```
    ************************************************************************
    *              Program Listing Section - PAYROLLX                      *
    ************************************************************************

000100        READ-A-RECORD.
000101            READ TIMECARD INTO WS-TIMECARD-REC
000102                AT END GO TO END-OF-JOB.
000103            MOVE WS-TIMECARD-REC TO MASTER-RECORD.
000104            ADD 1 TO WS-RECORD-CTR.
000105            IF TCR-EMP-NO > WS-EMP-NO
000106                MOVE TCR-EMP-NO TO WS-EMP-NO
000107            ELSE
000108                DISPLAY 'EMPLOYEE# SEQ ERROR  ',
000109                         TCR-EMP-NO,
000110                         SPACE,
000111                WS-EMP-NO UPON SYSOUT,
000112                GO TO READ-A-RECORD.
000113            MOVE TCR-EMP-NO TO RPT-EMP-NO.
000114            MOVE TCR-EMP-NAME TO RPT-EMP-NAME.
 CURR STMT ==>    SUBTRACT TCR-TOTAL-TAXES FROM TCR-GROSS-PAY
000116                GIVING WS-NET-PAY
000117            MOVE TCR-GROSS-PAY TO RPT-GROSS-PAY.
000118            MOVE TCR-TOTAL-TAXES TO RPT-TOT-TAXES.
000119            MOVE WS-NET-PAY TO RPT-NET-PAY.
000120            ADD WS-NET-PAY TO WS-TOTAL-NET-PAY.
```

Figure 12-26. Program Listing Section with PL/I XLS

```
    ************************************************************************
    *              Program Listing Section - PAYROLL                       *
    ************************************************************************


000059        CALC_HOURS: PROC;
000060        DCL     TOTAL_HOURS STATIC  FIXED DEC(5);
000061           REGULAR_TOTAL.EMP_NUMBER = OVERTIME_TOTAL.EMP_NUMBER =
                     EMPLOYEE_NUMBER_REMOTE_LOCATION;
000062           REGULAR_TOTAL.ACCRUED_HOURS = REGULAR_TOTAL.ACCRUED_HOURS +
                     REGULAR_HOURS_WEEKLY;
000063           OVERTIME_TOTAL.ACCRUED_HOURS = OVERTIME_TOTAL.ACCRUED_HOURS +
                     OVERTIME_HOURS_WEEKLY;
000064           TOTAL_HOURS = OVERTIME_TOTAL.ACCRUED_HOURS +
                              REGULAR_TOTAL.ACCRUED_HOURS;
 CURR STMT ==>    AVG_HOURS =  TOTAL_HOURS / MONTH_CODE;
000066            SORT_EMPL = MONTH || EMPLOYEE_NUMBER_REMOTE_LOCATION;
000067         END CALC_HOURS;
```

Figure 12-27. Program Listing Section with Assembler XLS

```
    ********************************************************************
    *              Program Listing Section - CWAADATE                  *
    ********************************************************************


   Stmt   Loc  Object Code       Source Statement
     42                        * Calculate years-of-service
     43                        * by subtracting employee year
     44                        * of hire from the current
     45                        * year.
     46                        ************************************************
     47                        CALC_ANNIVERSARY EQU *
     48                        * Uncomment following line before rolling
     49                        * this program into production!
     50                        *        ZAP   XYEARS,=P'3'
     51 00003C F221 B165 6000           PACK  PACKED_YEAR,RUN_YEAR
CURR => 000042 FD20 B165 B16A           DP    PACKED_YEAR,XYEARS
     53 000048 F900 B167 B176           CP    PACKED_YEAR+2(1),=P'0'
     54 00004E 4770 B064                BNE   EXIT
     55 000052 D501 7002 6002           CLC   HIRE_MONTH,RUN_MONTH
     56 000058 4770 B064                BNE   EXIT
     57 00005C 92E8 5000                MVI   ANNIV_IND,C'Y'
     58 000060 47F0 B064                B     EXIT
     60                        *********c************************************
     61                        * Exit ***************************************
     62                        ************************************************
```

File Section

The File section is a standard section that provides information for every file open at the time of the error. This section identifies:

- Data Management Control Block information
- DDNAME and dataset name
- File statistics
- Current and previous record information, when available
- Other information that is based on the type of file access method used.

File-AID Access

From the File section you can directly access a dataset through File-AID/MVS. Edit and Browse commands in the report take you into File-AID edit and browse modes within the dataset you've selected. A current version of File-AID/MVS must be available on the operating system.

The figure below shows a sample screen from which you access File-AID/MVS.

Figure 12-28. File Summary Selection List

```
COMPUWARE/VF  FILE SUMMARY SELECTION LIST  ------------------  ROW 1 TO 8 OF 8
COMMAND INPUT ===>                                          SCROLL ===> PAGE

  S - view file control blocks  E - File-AID Edit  B - File-AID Browse
                               (except JES spooled datasets)

  DDNAME  DEVICE UNIT VOLSER DISP AMTYPE  EXCPS   ------ DCB INFORMATION ---

  OUTLIST  JES2 spooled            QSAM   81       (DSORG=PS,RECFM=FBA,
                                                   LRECL=133,BLKSIZE=13300)
  MASTER   3380  3AB  CWX018 NEW   QSAM    0       (DSORG=PS,RECFM=FBA,
                                                   LRECL=80,BLKSIZE=3200)
S TIMECARD                         VSAM    1
  SYSUT1   3380  2C9  SMS801 SHR   VSAM    4
  SYSUT1   3380  2C9  SMS801 SHR   BDAM    4       (DOSORG=DA,RECFM=F,
                                                   LRECL=121,BLKSIZE=12100)
****************************** BOTTOM OF DATA *************************
```

Figure 12-29 shows a sample File section for an Abend-AID XLS report for a JES2 file.

Figure 12-29. File Section for a JES2 File

```
     Data Management Control Blocks for DDNAME - OUTLIST

JES2 Information:

Type.................Output
Line printer count...5
Line record outlim...999,999,999,999,999  (maximum allowed)
Byte count...........490  (including JES2 overhead)
DSNAME...............EFHJFGO.AA830JFG.JOB14558.D0000112.?

Program Environment:

Access method........QSAM
Unit address.........013B
VOL-SER..............SPOOL0
DCB Information......DSORG=PS,RECFM=FBA,LRECL=134,BLKSIZE=1340
Mode.................PUT LOCATE

Record Information:

Last record in JES2 spool

7F648200   CHAR    MICHAELS, ZACHARY E.                    02081256
           ZONE  44DCCCCCDE64ECCCCDE4C4444444444444444444FFFFFFFF44
          DIGIT  0049381532B0913819805B000000000000000000000208125600
                 1...5...10....*...20....*...30....*...40....*...50

7F648232   CHAR                          $ 2,324.69     $  123.09
           ZONE  444400000000044444444454F6FFF4FF44444544FFF4FF4444
          DIGIT  000000000000000000000B02B324B6900000B00123B090000
                51...*...60....*...70....*...80....*...90....*..100

7F648264   CHAR      $ 2,201.60
           ZONE  444454F6FFF4FF
          DIGIT  0000B02B201B60
               101...5...10....
Note: Trailing blanks suppressed by JES2.

Next to last record in JES2 spool

7F64818B   CHAR    TAYLOR, TIMOTHY T.                      01345678
           ZONE  44ECEDDD64ECDDECE4E4444444444444444444444FFFFFFFF44
          DIGIT  00318369B0394638803B000000000000000000000134567800
                 1...5...10....*...20....*...30....*...40....*...50

7F6481BD   CHAR                          $ 2,322.44     $   23.21
           ZONE  444400000000044444444454F6FFF4FF444445444FF4FF4444
          DIGIT  000000000000000000000B02B322B4400000B00023B210000
                51...*...60....*...70....*...80....*...90....*..100

7F6481EF   CHAR      $ 2,299.23
           ZONE  444454F6FFF4FF
          DIGIT  0000B02B299B23
               101...5...10....
```

Figure 12-30 shows a sample File section for a VSAM file.

Figure 12-30. File Section for a VSAM File

```
     Data Management Control Blocks for DDNAME - VDYN

     DSNAME=TEST.VSAM
                                                DISP=SHR   ACC METH=VSAM


File EXCP Count=4

SMSDATA:
  Data Class..........(Null)
  Storage Class.......NORMAL
  Management Class....TEST

File Summary:
  Access type.........BASE CLUSTER
  Dataset type........KSDS
  Processing type.....ADR,CNV,KEY,NFX,DSN,NDF,DIR,SEQ,NCI,IN,NIS,NRM,
                      NRS,NSR,NUB
  Max positions.......0000001
  Max active strings..0000002
  Key position........0000000
  Key length..........0000010

File errors...........None

Current key:
0A703B84   CHAR  8109928212
           ZONE  FFFFFFFFFF
          DIGIT  8109928212
                 1...5...10

Current record:
0A700F54   CHAR  8109928212JOHNSON,  KAREN.2  31  -        01200300
           ZONE  FFFFFFFFFFDDCDEDD644DCDCD4F44FF44644444444FFFFFFFF
          DIGIT  810992821216852650B0021955B2003100000000000001200300
                 1...5...10....*...20....*...30....*...40....*...50
0A700F86   CHAR  36012234324320004     00570000
           ZONE  FFFFFFFFFFFFFFFFF44444FFFFFFFF
          DIGIT  360122343243200040000000570000
                51...*...60....*...70....*...80

Previous record:
  No previous record for skip sequential or random processing.

Data Component:
  Number of buffers...0000000004
  High allocated RBA..000001B000 (hex)       110,592 (dec)
  High used RBA.......000001B000 (hex)       110,592 (dec)
  Available bytes.....0000073728
  CI splits...........0000000000
  CA splits...........0000000000
  Extents.............0000000001
  Max record length...0000000080
  Number of records...0000000321
  Records added.......0000000000
  Records deleted.....0000000000
  Records updated.....0000000201
  Records retrieved...0000000000

Index Component:
  Number of levels....0000000001
  Number of buffers...0000000002
  High allocated RBA..0000006200 (hex)        25,088 (dec)
  High used RBA.......0000000200 (hex)           512 (dec)
  Available bytes.....0000024576
  Extents.............0000000001
  Number of records...0000000001
  Records updated.....0000000000

Last request:
  Description........ Keyed mode
  ID..................GET
  Request ECB.........Completed
  Mode................MOVE
  Access..............SEQ
  Misc................Locate the record identified by a key
  Current RBA.........00003E30 (hex)        15,920 (dec)
```

Figure 12-31 shows a sample File section for a QSAM file.

Figure 12-31. File Section for a QSAM File

```
     Data Management Control Blocks for DDNAME - MASTER

     DSNAME=SYS96297.T104522.RA000.AA830JFG.TEMP.H01

 Device=3390   Unit ADR=VIO                    DISP=NEW   ACC METH=QSAM

File EXCP Count=0                             Mode=PUT LOCATE
             DCB=(DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=3200)


DCB

00000 00006CA0  09BF0038 00000000 00000001 00F3508C  .............3&.
00010 00006CB0  002FE5A2 05048090 00004000 00005968  ..Vs...... .....
00020 00006CC0  C6000001 900463BC 00F40048 008B84F4  F........4....d4
00030 00006CD0  92E746A0 00CAAB58 0A04CE70 02090C80  kX.............
00040 00006CE0  30013030 000059D8 09BF8E00 09BF82C0  .......Q......b{
00050 00006CF0  00000050 00000000 00000000           ...&........


     Data Management Control Blocks for DDNAME - MASTER


Previous record

09BF81D0   CHAR  0208125602MICHAELS, ZACHARY E.            02324690
           ZONE  FFFFFFFFFFDCCCCCDE64ECCCCDE4C4444444444444FFFFFFFF
          DIGIT  0208125602493815320913819805B0000000000000002324690
                 1...5...10....*...20....*...30....*...40....*...50

09BF8202   CHAR  1230923423423
           ZONE  FFFFFFFFFFFFF44444444444444444
          DIGIT  1230923423423000000000000000000
                51...*...60....*...70....*...80

Last record written

09BF8220   CHAR  0307169102ROBERTSON, KENDALL A.             1203332
           ZONE  FFFFFFFFFFDDCCDEEDD64DCDCCDD4C4444444444444FFFFFFF
          DIGIT  0307169102962593265B0255413301B0000000000001203332
                 1...5...10....*...20....*...30....*...40....*...50

09BF8252   CHAR  31299223442344
           ZONE  FFFFFFFFFFFFFF4444444444444444
          DIGIT  3129922344234400000000000000000
                51...*...60....*...70....*...80

Current record area

09BF8270   CHAR  0406029302EDWARDS, ALEXANDER G.           @@331100
           ZONE  FFFFFFFFFFCCECDCE64CDCECDCCD4C44444444444477FFFFFF
          DIGIT  0406029302546194 2B013571545907B000000000000CC331100
                 1...5...10....*...20....*...30....*...40....*...50

09BF82A2   CHAR  1322243234324
           ZONE  FFFFFFFFFFFFF44444444444444444
          DIGIT  1322243234324000000000000000000
                51...*...60....*...70....*...80
```

Note: DEB, IOB, and UCB information is available for a QSAM file, but is not shown in this figure.

Figure 12-32 shows a sample File section for an IAM file.

Figure 12-32. File Section for a IAM File

```
     Data Management Control Blocks for DDNAME - SYSUT1
     DSNAME=AA.TECH.SAMPLES.IAM.KSDS
 Device=3390   Unit ADR=02A6  VOL-SER=PRD926   DISP=SHR   ACC METH=VSAM

File EXCP Count=4

SMSDATA:
   Data Class........(Null)
   Storage Class.....STDNOCSH
   Management Class..@NONSTD@
Access method.........IAM file access

File Summary:
  Access type.........BASE CLUSTER
  Dataset type........KSDS
  Processing type.....KEY,NFX,DDN,NDF,SEQ,NCI,IN,NIS,NRM,NRS,NSR,NUB
  Max positions.......0000001
  Max active strings..0000000
  Key position........0000000
  Key length..........0000004
File errors...........None

Current key:
00005874   CHAR  1001
           ZONE  FFFF
          DIGIT  1001
                 1...

Current record:
00007980   CHAR  1001 NEW JERSEY
           ZONE  FFFF4DCE4DCDECE4444444444444444444440010000000000000
          DIGIT  1001055601592580000000000000000000000C00C00C00C00C
                 1...5...10....*...20....*...30....*...40....*...50
000079B2   CHAR
           ZONE  00000000000000000000000000000018
          DIGIT  00C00C00C00C00C00C00C00C00C03F
                51...*...60....*...70....*...80
Previous record not available for IAM files.

Data Component:
  Number of buffers...0000000012
  High allocated RBA..00000E3000 (hex)       929,792 (dec)
  High used RBA.......0000030000 (hex)       196,608 (dec)
  Available bytes.....0000735300
  CI splits...........0000000000
  CA splits...........0000000000
  Extents.............0000000000
  Max record length...0000004081
  Number of records...0000000004
  Records added.......0000000000
  Records deleted.....0000000000
  Records updated.....0000000000
  Records retrieved...0000000000

Index Component:
  Number of levels....0000000000
  Number of buffers...0000000012
  High allocated RBA..00000B3000 (hex)       733,184 (dec)
  High used RBA.......00000B3000 (hex)       733,184 (dec)
  Available bytes.....0000735300
  Extents.............0000000000
  Number of records...0000000004
  Records updated.....0000000000

Last request:
  Description........ Last READ was NOTE STRING position
  ID..................GET
  Request ECB.........completed
  Mode................MOVE
  Access..............SEQ
  Misc................Locate the record identified by a key
  Current RBA.........00000000 (hex)             0 (dec)
```

DB2 Section

Refer to “Abend-AID for DB2” on page 7-1 for a complete description.

IMS Section

Refer to "Abend-AID for IMS" on page 7-11 for a complete description.

IDMS Section

Refer to "Abend-AID for IDMS" on page 7-15 for a complete description.

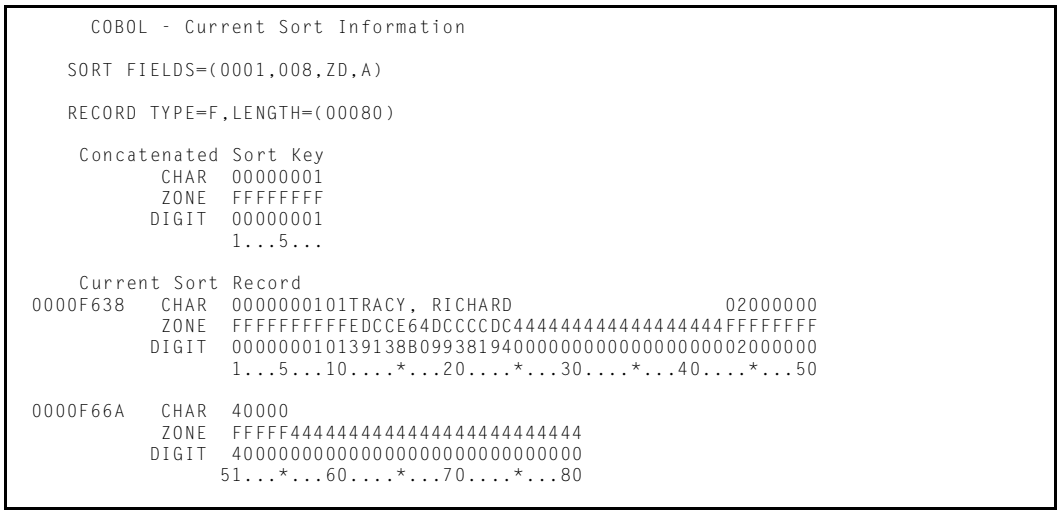
Current Sort Record Section

The Current Sort Record section is available only when the program uses an internal COBOL sort. This section identifies:

- Sort fields. For example: SORT FIELDS=(0001,008,ZD,A)
- Record type and record length. For example: Record Type=F,Length=(00080)
- Concatenated sort key
- Current sort record in vertical-hexadecimal format.

Figure 12-33 shows a sample Current Sort Record section for a COBOL program.

Figure 12-33. Current Sort Record Section



```
      COBOL - Current Sort Information

    SORT FIELDS=(0001,008,ZD,A)

    RECORD TYPE=F,LENGTH=(00080)

     Concatenated Sort Key
            CHAR  00000001
            ZONE  FFFFFFFF
           DIGIT  00000001
                  1...5...

     Current Sort Record
0000F638    CHAR  0000000101TRACY, RICHARD                  02000000
            ZONE  FFFFFFFFFFEDCCE64DCCCCDC44444444444444444444FFFFFFFF
           DIGIT  000000010139138B09938194000000000000000000002000000
                  1...5...10....*...20....*...30....*...40....*...50

0000F66A    CHAR  40000
            ZONE  FFFFF4444444444444444444444444
           DIGIT  40000000000000000000000000000000
                 51...*...60....*...70....*...80
```

COBOL Data Locator

Presents a summary of all data-related source and formats in an abended COBOL program, identifying formats and data that may be invalid.

The locator search specifications can be changed for any dates or data you need by entering the FINDDATA command on its command line. The summary will be automatically refreshed with the specified data when you exit.

Figure 12-34. COBOL Data Locator Summary

```
COMPUWARE/VF ----------------------------------------------------------------
COMMAND INPUT ===> _                                        SCROLL ===> CSR
                                                  Formatting has completed
'PFHWFBO.DUMP.DDIO'      REPT.71   PFHWFBV2  J06619
 Program: MLECOB          Compile Date: 07 JAN 2000    YEARWINDOW(1971)

  '@' in column    36 denotes field with Data Locator PICTURE match.
  '!' in column    36 denotes field with Data Locator DATE    match.
  '#' in column    36 denotes field contains data not matching PICTURE clause.

    Language Processor Interface Release      = 07.07.00
    Language Processor Post Processor Release = 07.07.00


             Working-Storage Section

Level/Field Name                Picture/Type Offs        Value
------------------------------  -----------------   ----+----1----+----2
77 CURRENT-YEAR               #  99                  SPACES
77 DAT-DATA-MATCH             !  9(5)                99366
77 DUE-YEAR                      99                  05
              DATE FORMAT    YY       YEAR(2005)
77 END-YEAR                      9999                2005
              DATE FORMAT  YYYY       YEAR(2005)
77 EXP-CUST-DATE              @  9(8)                19580430
              DATE FORMAT  YYYYXXXX   YEAR(1958)
01 BASE-DATE                                        580430
              DATE FORMAT    YYXXXX   YEAR(2058)
02 BASE-YEAR                     9(2)                58
02 BASE-MONTH                    9(2)                04
02 BASE-DAY                      9(2)                30
```

Abend-AID for WebSphere MQ Section

Displays either MQSeries batch or MQSeries IMS information created by Abend-AID for WebSphere MQ.

Figure 12-35. MQSeries Selection List

```
COMPUWARE/VF ----- WebSphere for MQ Selection List----------- Row 1 to 2 of 2
COMMAND INPUT ===>                                          SCROLL ===> PAGE

 S - view WebSphere for MQ Information

  Name     Description

  MQBatch  WebSphere for MQ batch information
  MQIMS    WebSphere for MQ IMS information
****************************** Bottom of data ******************************
```

For more information about Abend-AID for WebSphere MQ, refer to the *Abend-AID for WebSphere MQ Installation and Product Description Manual*.

LE Section

Abend-AID XLS collects and interprets the runtime options from LE's options control block. This information is available from Abend-AID XLS's Language Environment Selection List.

Figure 12-36. LE information identifying LE run-time options (OCB)

```
*************************************************************************
* OS/390 V2R8 LE      Options in effect at time of error                *
*************************************************************************

OPTION
TABLE  in  Module     Loaded From
-------------------------------------------------------------------------
CEEDOPT    CEEBINIT   CEE.SCEERUN

 OPTION                                        SET FROM
-------------------------------------------------------------------------
  ABPERC(NONE)                                 CEEDOPT
  ABTERMENC(ABEND)                             INVOCATION (JCL,PARMS ETC)
NOAIXBLD                                       CEEDOPT
  ALL31(OFF)                                   CEEDOPT
  ANYHEAP(16384,8192,ANYWHERE,FREE)            CEEDOPT
NOAUTOTASK                                     CEEDOPT
  BELOWHEAP(8192,4096,FREE)                    CEEDOPT
  CBLOPTS(ON)                                  CEEDOPT
  CBLPSHPOP(ON)                                CEEDOPT
  CBLQDA(ON)                                   CEEDOPT
  CHECK(ON)                                    CEEDOPT
  COUNTRY(US)                                  CEEDOPT
  DEBUG                                        CEEDOPT
  DEPTHCONDLMT(10)                             CEEDOPT
  ENVAR("")                                    CEEDOPT
  ERRCOUNT(20)                                 CEEDOPT
  ERRUNIT(6)                                   CEEDOPT
   FILEHIST)                                   CEEDOPT
```

LE Control Blocks

Abend-AID XLS displays significant control blocks from the LE runtime environment. As an example, Figure 12-37 on page 12-27 shows the Common Anchor Area.

Figure 12-37. LE Section – Common Anchor Area

```
                    Significant Fields in CEECAA Control Block

   CEECAA:  00019A38
   +0002B0  CEECAALEVEL              LE/370 level indicator
   +0002D8  CEECAAERR                A(Current CIB)
   +0002E0  CEECAADDSA               A(Dummy DSA)
   +0002F0  CEECAAEDB                A(EDB)
   +0002F4  CEECAAPCB                A(PCB)
   +000350  CEECAARCB                A(RCB)

   ************************************************************************
   *  Dump of CEECAA  -  Common Anchor Area                               *
   *                     Each thread is represented by a CAA              *
   ************************************************************************

 DSPL Address   -------------- Data ---------------
00000 00019A38  00000800 00000000 0002A000 00000000  ................

00010 00019A48  00000000 00000000 00000000 00000000  ................
            LINES 00019A58-00019A98 SAME AS ABOVE
 DSPL Address   -------------- Data ---------------

00070 00019AA8  00000000 80016690 00000000 00000000  ................
00080 00019AB8  00000000 00000000 00000000 00000000  ................
            LINES 00019AC8-00019B48 SAME AS ABOVE
00120 00019B58  100125E0 00000000 00000000 00000000  ...\............
00130 00019B68  00000000 00000000 00000000 00000000  ................
            LINES 00019B78-00019BB8 SAME AS ABOVE
00190 00019BC8  00000000 00000000 50C0D064 05C058C0  ........&{}..{.{
001A0 00019BD8  C00605CC 000121AE 0700C198 0700C198  {.........Aq..Aq
001B0 00019BE8  0700C198 0700C198 0700C198 0700C198  ..Aq..Aq..Aq..Aq
            LINES 00019BF8-00019C18 SAME AS ABOVE
001F0 00019C28  00000000 00000000 00000000 00000000  ................
            LINES 00019C38-00019C78 SAME AS ABOVE
00250 00019C88  00000000 00000000 00000000 10011F30  ................

00260 00019C98  00000000 00000000 80013888 80013578  ...........h....
00270 00019CA8  00000000 00000000 00000000 00000000  ................
            LINES 00019CB8-00019CC8 SAME AS ABOVE
002A0 00019CD8  00000000 00000000 00000000 03030290  ................
 DSPL Address   -------------- Data ---------------

002B0 00019CE8  09000000 00015F28 0001F038 00015E38  ......¬...0...;.
002C0 00019CF8  00028000 00000000 00028018 100C4F9E  ..............|.
002D0 00019D08  84000FCA 00000000 10013AE8 00015BB8  d.........Y..$.
002E0 00019D18  0001A2E0 00000000 00000000 00000000  ..s\............
002F0 00019D28  000189C8 00018558 00019A20 00019A38  ..iH..e.........
00300 00019D38  00015AD0 00000000 00000000 00000000  ..!}............
00310 00019D48  00000000 00000000 00000000 00016018  ..............-.
00320 00019D58  00000000 00000000 00000000 00000000  ................
00330 00019D68  00000000 00000000 00000000 80000000  ................
00340 00019D78  00000000 00000000 00000000 00000000  ................
00350 00019D88  00017918 00000000 0001A380 00000000  ..`.......t.....
00360 00019D98  00000000 00000000 00000000 00000000  ................
            LINES 00019DA8-00019DB8 SAME AS ABOVE
00390 00019DC8  00019DCC 00000001 00000000 F8000000  ............8...
003A0 00019DD8  84000000 0000000D 84000FCA 00000000  d.......d.......
003B0 00019DE8  0000DDB0 00000000 10051490 00000000  ................
003C0 00019DF8  00000000 00000000 00000000 00000000  ................
****************************** Bottom of data ******************************
```

LE Heap Storage

Abend-AID XLS validates and analyzes LE Heap Storage. If damage to the heap storage structure is detected, Abend-AID XLS provides diagnostic information to assist in problem determination and resolution. This includes valuable storage displays to help determine the cause of the original error.

Figure 12-38 on page 12-28 shows the LE Heap Storage Report section when damage to the heap storage structure is detected.

Figure 12-38. LE Section – Heap Storage – with errors

```
                              LE Heap Storage

  User Heap:

  Initial size.............................    32768
  Increment size...........................    32768
  Number of segments allocated............        1

  ***********************************************
  *     Heap Header for Segment at 1011A000:    *
  ***********************************************
  Heap ID..................................        0
  Segment address..........................  1011A000
  Segment length...........................    32768
  Root address (largest free element).....  1011A038
  Root length..............................    32712

  Allocation Map for Segment at 1011A000:

  Element Addr    Data Addr    Data Len     Status
  ------------    ---------    --------     ------
  1011A020        1011A028           16     Alloc
  1011A038        1011A048        32696     Free     ** Error

  ** ERROR:

  Left pointer in free element header at 1011A038 is invalid.

  Possible causes:

  The header of the free element may have been overlayed.
  located at 1011A020. The element is large enough to
  hold 00000016 (X'00000010') bytes of user data.

  The allocated storage begins at 1011A028 and contains:
  DSPL Address   --------------- Data ---------------
  00000 1011A028  F1F2F3F4 F5F6F7F8 F9F0F1F2 F3F4F5F6  1234567890123456

  The content of the damaged free element header at 1011A038,
  and surrounding data is:
  DSPL Address   --------------- Data ---------------
  00000 10119FF8  00000000 00000000 C8C1D5C3 00017E60  ........HANC..=-
  00010 1011A008  00017E60 00000000 1011A000 1011A038  ..=-............
  00020 1011A018  00008000 00007FC8 1011A000 00000018  ......"H........
  00030 1011A028  F1F2F3F4 F5F6F7F8 F9F0F1F2 F3F4F5F6  1234567890123456
  00040 1011A038  F7000000 00000000 00000000 00000000  7...............

  DSPL Address   --------------- Data ---------------
  000050 1011A048  00000000 00000000 00000000 00000000  ................
            LINES 1011A058-1011A068 SAME AS ABOVE
  00080 1011A078  00000000 00000000 00000000 00000000  ................

  User Action:

  Review storage allocations and use in the application.

  Summary of Analysis for Segment at 1011A000:

  Number of allocated elements.............       1
  User storage allocated (in bytes).......      16
  Number of free elements..................       1
  User storage available (in bytes).......   32696
   Number of errors........................       1

   Heap storage validation complete -- 001 error detected.
 ******************************** Bottom of Data ****************************
```

Figure 12-39 shows the LE Heap Storage Report section when no errors were detected in Heap Storage.

Figure 12-39. LE Section – no errors

```
                        LE Heap Storage

User Heap:

Initial size.....................................       32768
Increment size...................................       32768
Number of segments allocated.....................           1

************************************************************
*              Heap Header for Segment at 1011A000:        *
************************************************************
Heap ID..........................................           0
Segment address..................................    1011A000
Segment length...................................       32768
Root address (largest free element)..............    1011A000
Root length......................................       32664


Allocation Map for Segment at 1011A000:

Element Addr          Data Addr       Data Len       Status
------------          ---------       --------       ------
1011A020              1011A028              16       Alloc
1011A038              1011A040               8       Alloc
1011A068              1011A078           32648       Free
1011A048              1011A050               8       Free
1011A050              1011A058              16       Alloc


Summary of Analysis for Segment at 1011A000:

Number of allocated elements.....................           3
User storage allocated (in bytes)................          40
Number of free elements..........................           2
User storage available (in bytes)................       32656


Heap storage validation complete -- no errors detected.
```

Runtime Library Services (RTLS)

Figure 12-40. LE Section – RTLS Information

```
COMPUWARE/VF ------------------------------------------------ Row 1 to 2 of 2
COMMAND INPUT ===>                                          SCROLL ===> PAGE

'AA.R940.ENHINT.RPTFILE'  REPT.65  AA940000  J23104

          RTLS - ON / ACTIVE  LIBRARY - SCEERUN   VERSION - CURRENT

****************************** Bottom of data ******************************
```

As shown in Figure 12-40, Abend-AID XLS identifies the RTLS library name and version in use at the time of the error.

Epilog Section

The Epilog section, at the end of a report, summarizes Abend-AID XLS's action.

- How the report was printed, if applicable:
  - From batch (as a SYSOUT of your JOB)
  - From an Abend-AID XLS report dataset, identifying name and number.
- The vertical and horizontal hexadecimal translation tables.
- Whether a system dump was printed or suppressed.
- The following warning messages when applicable:

```
The IBM COBOL Load List for this application appears to be corrupted.

The system save area chain for this application appears to be broken.

The COBOL environment appears to be corrupted.

Working Storage Display was suppressed for one or more CSECTs by
CSECTBYP.

CWINCLUD processing was suppressed because the CWINCLUD table is
incompatible with this release of Abend-AID XLS. Contact your Abend-AID
XLS installer to run JCLINCLD.
```

Report Routing Information

- Abend-AID XLS's memory utilization statistics. Note that the Above, Below, and Total figures refer to the largest amount ever in use for each category. The amounts shown in the Total lines will tend to, but not necessarily, agree with the totals of the Above and Below detail lines.
- How Abend-AID XLS was called.

Figure 12-41. Epilog Section

```
************************************************
* This software is confidential and proprietary  *
* to Compuware Corporation. No use or disclosure *
* is permitted other than as expressly set forth *
* by written license with Compuware Corporation. *
*------------------------------------------------*
*      Report printed from Abend-AID batch       *
*  Default table used to translate vertical hex  *
* Default table used to translate horizontal hex *
*------------------------------------------------*
*   No dump was requested by installations table *
*  Table CWTABS01 loaded from JOB/STEPLIB+ 001   *
*   The ID of the suppressed dump is 000         *
*------------------------------------------------*
* Report Routing Information - overrides listed  *
*  in order of precedence, highest first:        *
*                                                *
* >Requested via //ABNLTERM DD:                  *
*  SYSOUT                                        *
*                                                *
* >Requested via CWROUTE site default:           *
*  AA.PROD.RPTSHDIR                              *
*                                                *
* CWROUTE was loaded from STEPLIB library:       *
*  COMPWARE.CUST.LOAD                            *
* It was link edited on 26 JUL 2002 and is       *
* 00000170 bytes long                            *
*------------------------------------------------*
* Maximum Abend-AID Memory Utilization (bytes):  *
*  Above 16M - LOAD:  88.36K   GETMAIN:  58.08K  *
*  Below 16M - LOAD:  60.58K   GETMAIN:  46.88K  *
*           Above 16M - Total: 143.56K           *
*           Below 16M - Total: 107.46K           *
*------------------------------------------------*
*     Abend-AID gained control through LEAID     *
************************************************
```

Figure 12-42. Epilog Section with Warning

```
                        *** WARNING ***

This Abend-AID report may be incomplete for the following reason(s):

   - CWINCLUD processing was suppressed because the CWINCLUD table
     is incompatible with this release of Abend-AID.  Contact your
     Abend-AID installer to run JCLINCLD.

         ************************************************
         *      Report printed from Abend-AID batch       *
         *  Default table used to translate vertical hex  *
         * Default table used to translate horizontal hex *
         *----------------------------------------------*
         *    Dump requested due to Warning Message       *
         ************************************************
```

Chapter 13.
Understanding the Compuware/VF Interface

This chapter summarizes the use of Abend-AID XLS screens, field by field.

Compuware/VF Screens

Use Compuware/VF (Viewing Facility) screens to interactively view Abend-AID reports and Compuware source listings in an ISPF/PDF-like environment.

Entry Panel

Figure 13-1 shows the Compuware/VF Entry Panel, which you obtain when you access Abend-AID XLS. Here you specify the datasets you want to use for XLS or base language support. For datasets on remote MVS images, you also specify the Abend-AID XLS server identifications that you need for distributed viewing. You can also set processing conditions.

Figure 13-1. Entry Panel

```
COMPUWARE/VF ENTRY PANEL --------- COPYRIGHT COMPUWARE CORPORATION 1976, 2002.
Command ===>
Dataset choice . 0        (Select 0 - 8, from below, or C for Contact Data)
Member . . . . . ________ (MVS:jobname, SOURCE:program, or member*)
                          (Use the LNAME command to enter a Long Program Name)
                                                                  Remote
       Abend-AID Shared Directory or Report Dataset:         Volume  Server
           0 'AA.REPORTS'
       Source Directory or Listing Datasets:
           1 'AA.LISTING'
           2
           3
           4
           5
           6
           7
           8
Processing options:                                  Source browsing options:
  Confirm delete  . .  YES      (Yes or no)             Unit . . . . . VIO
  Show print setup  .  YES      (Yes or no)             Blocking . . . 10
  Show source warning  YES      (Yes or no, used only for Abend-AID)
  Language  . . . . .  ENGLISH
  Local Server SSID .           (Required for Viewing Remote DDIO files)
```

The following sections discuss possible ways to complete each field on the Entry Panel. Your input is saved for future sessions.

Command Prompt

Enter one of the following:

- Valid TSO command
- CSSUTIL command to invoke CSS Utilities
- LLNAME command to enter a long program name or wildcard names to be used for member search. See Member description below.

Note: Refer to the *Compuware Shared Services User/Reference Guide* for a complete description of the CSS Utilities.

Dataset Choice

Enter the number corresponding to a report dataset or source listing dataset listed in the Abend-AID Shared Directory or Report File or Source Directory or Listing Files section of this screen. Or enter C for contact information

Enter one of the following based on the type of output you desire:

- To view an Abend-AID XLS error analysis report, enter 0 in the Dataset choice field.
- To view only a Source Listing, enter 1 through 8 in the Dataset choice field.
- To view only a Source Listing stored in a dataset not currently available in the list, enter a number that is not currently being used (for example, in Figure 13-1 on page 13-1 numbers 2 through 8 are not used). When you enter a number not currently being used, you must also enter the source listing dataset name to the right of the corresponding number in the list.

  Notes:

  1. If a dataset number is not specified, 0 will be used.
  2. When you press Enter after completing the remaining fields on the Compuware/VF Entry Panel, the Dataset Directory screen is displayed. For a description of this screen, see "Dataset Directory" on page 13-4.

- To display contact information, enter C in the Dataset choice field.

  The Contact Information feature allows you to associate contact information (for example, the phone number and e-mail address of the off-duty systems programmer) with a particular job name. If a job abends, users may search the contact information list to determine who to inform about the abend.

Member

Specify the member name or group of member names of the file selected via your entry in the Dataset Choice field. Enter one of the following:

blank	Lists all members of the dataset choice above.
job name or program name	Lists all members in the file associated with the job name (for Abend-AID reports) or program name (for source listings).
Member*	Lists all members in the file with the job or program names beginning with the characters specified before the wildcard character '*'. To enter a program name or wildcard longer than eight characters, you must use the LNAME command **instead**. The LNAME command is described in "Command Prompt" on page 13-1.

Abend-AID Shared Directory or Report File

To work with Abend-AID reports, enter the name of the Abend-AID report shared directory, report database, or report file. The dataset entered in this field (corresponding to Dataset choice 0) is the primary dataset. If the high-level qualifier of the dataset name is your TSO ID (prefix), begin with the next qualifier; otherwise, enclose the dataset name in single quotation marks.

If you do not use Abend-AID, you may specify a source listing file in this entry (the 0 entry).

Source Directory or Listing Files

Enter the names of one or more source listing shared directories, source listing databases, or listing files containing the output of the Compuware COBOL, PL/I, Assembler, or C language processor. If the high-level qualifier of the dataset name(s) is your TSO ID (prefix), begin with the next qualifier. Otherwise, enclose the dataset name(s) in single quotation marks.

This area lists up to eight source listing datasets on lines 1 - 8. To view one of the listed datasets, the line number appearing to the left of the dataset name must be entered in the Dataset choice field.

The number on the line determines the order in which the datasets are searched when merging source information with an Abend-AID XLS report. Blank lines can occur anywhere in the list.

Volume

If the report or source listing dataset is not cataloged, or the catalog is not available, enter the serial number of the DASD volume containing the dataset.

Remote Server

If a dataset is located on a remote MVS system, identify the Abend-AID server on that system. Both the local and remote Abend-AID servers must be configured to use Compuware Base Services/HCI, which is required to support Distributed Viewing Support.

Note: Not currently available for source listing databases or shared directories.

Processing Options

Use these options to determine the format of the output, and to display or suppress certain information.

Confirm delete

Enter **YES** to display a Confirm Delete screen before deleting a report or source listing.

Show print setup

Enter **YES** to display and update print options prior to printing reports or source listings. Refer to "Hardcopy Options" on page 13-8 for additional information.

Show source warning

Enter **YES** to display a warning screen when the compile date and time of the program in the Abend-AID report does not match a source listing in the source listing dataset. Refer to "COBOL Mismatched Date and Time" on page 2-14 for additional information.

Language

The default is mixed-case English. Enter **USAUC** for uppercase English if your terminal does not support mixed case. Enter **JAPANESE** for terminals that support DBCS.

Local Server SSID

If you are accessing a dataset on a remote MVS system, identify the Abend-AID XLS server of the local system. Both the local and remote Abend-AID XLS servers must be configured for distributed viewing.

Source browsing options

These options create a temporary file that is used when browsing a source listing.

Unit

Enter the type of device where the temporary file is to be allocated when browsing a source listing. Recommended entries are **VIO** (if site-supported) or **SYSDA**.

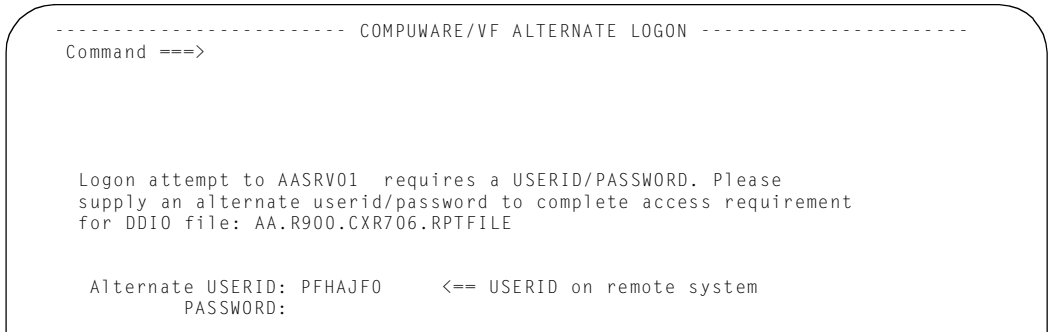
Blocking

Enter the number of records in each block of the temporary file. A larger number may improve response time. If you want Abend-AID XLS to calculate the most efficient blocking for you, enter **9999**. Abend-AID XLS sets the blocking after you view a source listing.

Alternate Logon

If you entered specifications at the Entry Panel for access to a remote MVS system, the Compuware/VF displays the Alternate Logon screen. Here you log on to the Abend-AID XLS server on that system.

Figure 13-2. Alternate Logon



Alternate USERID, PASSWORD

Enter your ID and password to access the remote Abend-AID XLS server.

Dataset Directory

The dataset directories show the members in the datasets specified in the Report dataset and Source dataset fields of the Compuware/VF Entry Panel.

Abend-AID Dataset Directory

The Abend-AID Dataset Directory screen displays the list of members in the file specified in the Report dataset field of the Entry Panel. The Abend-AID Dataset Directory screen for a report shared directory is shown in Figure 13-3 on page 13-5. By default, the members are listed in order according to date and time on the Entry Panel.

Figure 13-3. Abend-AID Dataset Directory

```
COMPUWARE/VF ---- 'ABENDAID.REPORT.FILE'--------------------------  ROW 1 OF 52
COMMAND INPUT ===>                                                SCROLL ===> PAGE
                        ABEND-AID DATASET DIRECTORY

      S - View C - Contact Info D - Delete P - Print L - Lock U - Unlock

   JOBNAME   REPT NUMBER JESID   CODE       DATE       TIME    DESC       SIZE(K)
  TSS070CS           71  J08263  SOC7  05 MAR 2000  17.50.22 CLAYTON D     31
  TSS070CS (L)       70  J07553  U1035 05 MAR 2000  16.45.18 DONNA C       15
  TSM050TS           68  J09691  U0240 02 MAR 2000   9.11.49 WOODS         15
  TSM050TS (L)       67  J09465  SOC2  02 MAR 2000   8.51.20 T WOODS       15
  TSM030SN           66  J05446  SNAP  01 MAR 2000  16.33.07 T SMITH       15
  TSM020TX           65  J00942  SOC1  01 MAR 2000   9.31.15 GARY K        15
  TSM050TS           64  J00406  U1020 01 MAR 2000   8.42.12 WOODS TINA    15
  TSM050TS           63  J07030  SOC7  10 FEB 2000  16.54.14 KAYLOR G      31
  TSM050TS (L)       62  J06927  SOCB  10 FEB 2000  16.46.36 WOODS T       31
  TSM030LP           61  J05611  SOC7  10 FEB 2000  15.07.38 SMITH         15
  TSM030PL           59  J09984  S213  10 FEB 2000  14.50.09 T SMITH       15
  TSM050X1 (M)       58  J08064  SD37  08 FEB 2000  11.29.39 TINA WOODS    15
  TSS070CS           57  J08885  SOC7  05 FEB 2000  16.53.20 DONNA C       31
  TSA060RI           55  J00512  U2222 03 FEB 2000  13.54.07 RANDY S       31
  TSA060RO           52  J08065  SOC7  02 FEB 2000  17.36.52 SMITH R       31
  TSM020ST           51  J03476  SOCF  02 FEB 2000  10.19.17 KAYLOR G      15
  TSA060RA           50  J09305  U2222 01 FEB 2000  15.41.46 RANDY S       31
  TSA080PR           46  J05500  SOC7  01 FEB 2000   9.19.48 HEIDI M       15
  TST020LB           45  J06929  SD37  31 JAN 2000  12.52.31 FRANKEN S     15
```

Each report dataset directory entry contains the following:

JOBNAME

The job name.

REPT NUMBER

The number assigned by Abend-AID to identify the report. The following status indicators also appear if applicable:

- **(L):** Report is automatically locked.
- **(M):** Report is manually locked.
- ***I:** Report is incomplete.
- ***IL:** Report is incomplete and automatically locked.
- ***IM:** Report is incomplete and manually locked.

JESID

The job number assigned by the Job Entry Subsystem.

CODE

The system or user abnormal termination code, or “SNAP” for a SNAP-AID report, or the PL/I on-code.

DATE

The date on which the error occurred or the date of the SNAP-AID report.

TIME

The time at which the Abend-AID or Snap-AID report was created.

DESC

By default, the first ten characters of the PROGRAMMER NAME field of the job statement. When using Abend-AID user exit CWEXIT02 or when calling Snap-AID, you can override this value.

SIZE

The number of bytes allocated for the report in the report dataset.

Source Listing Dataset Directory

The Abend-AID Source Listing Dataset Directory screen displays all source listings in a specified source listing dataset.

Figure 13-4. Source Listing Dataset Directory

```
COMPUWARE/VF -- Source Listing Standard DDIO Member List --- Row 1 to 14 of 67
COMMAND INPUT ===>                                          SCROLL ===> PAGE
Source DDIO File: AA.R940.ENHINT.LSTFILE



SELECT S - Select   P - Print   D - Delete   L - Lock   U - Unlock   I - Info
CODE   PROGRAM     LIST NUMBER     COMP DATE     TIME     RC   LANGUAGE  SIZE(K)
______ ABNDIMM   (L)        67    31 MAY 2000  13:08:27  04   COBOL/390M     48
______ ABNDIMM             66    31 MAY 2000  13:06:57  04   COBOL/390M     48
______ ABEND476  (L)        65    30 MAY 2000  16:06:38  04   COBOL/MVSM     80
______ TEST400Z  (L)        64    30 MAY 2000  15:35:29  04   COBIIR4        32
______ TESTSRTA  (L)        63    30 MAY 2000  15:35:13  04   COBIIR4        64
______ TEST40LE  (L)        62    30 MAY 2000  14:16:29  04   COBOL/MVSM     48
______ IMSSRTLE  (L)        61    30 MAY 2000  14:16:20  04   COBOL/MVSM     48
______ TEST400B  (L)        60    30 MAY 2000  14:14:24  04   COBOL/390M     48
______ TESTSRTB  (L)        59    30 MAY 2000  14:14:08  04   COBOL/390M     80
______ CWAASUBC  (L)        58    30 MAY 2000  13:51:30  00   COBOL/370      32
______ CWAAHOUR  (L)        57    30 MAY 2000  13:51:15  00   COBOL/370      32
______ CWAADATE  (L)        56    30 MAY 2000  13:51:00  00   COBOL/370      32
______ CWAACOB   (L)        55    30 MAY 2000  13:50:43  00   COBOL/370     112
______ CWAASUBC             54    30 MAY 2000  13:49:10  00   COBOL/370      32
```

Each source listing directory entry displays the following information:

PROGRAM

The program name.

LIST NUMBER

The number assigned by the Compuware language processor to identify the source listing. The following status indicators also appear if applicable:

- **(L):** Source listing is automatically locked.
- **(M):** Source listing is manually locked.
- ***I:** Source listing is incomplete.
- ***IL:** Source listing is incomplete and automatically locked.
- ***IM:** Source listing is incomplete and manually locked.

COMP DATE

The date the program was compiled.

TIME

The time the program was compiled.

RC

The Compuware language processor return code.

LANGUAGE

The program language.

SIZE

The number of bytes allocated for the source listing in the source listing dataset.

Primary Commands

In addition to the standard scrolling commands, you can enter the following commands in the COMMAND INPUT area on the Dataset Directory screens:

LOCATE (L) jobname

Positions the matched job name as the first entry on the Abend-AID Dataset Directory screen. The asterisk (*) wildcard character is valid.

LOCATE (L) program

Positions the matched program name as the first entry on the Source Listing Dataset Directory screen. The asterisk (*) wildcard character is valid.

NEXT (N) nnnn

Scrolls forward to the job or program name specified in the previous LOCATE command, where *nnnn* is the next nth match to be located. The nnn default value is **1**.

SORT

Sorts the directory list by the specified sort key. Sort-key can be any column except TIME. By default, the directory is displayed in descending order by REPT NUMBER for Abend-AID reports and LIST NUMBER for source listings.

- JOBNAME (JN)
- NUMBER (NUM)
- JESID
- CODE
- DATE
- DESC
- SIZE

Valid sort keys and their abbreviations for the source listing directory are:

- PROGRAM (PGM)
- NUMBER (NUM)
- DATE
- RC
- LANGUAGE (LANG)
- SIZE

Line Commands

The following line commands are also valid on the Dataset Directory screens. Enter one or more of the following commands to the left of the desired report or source listing directory entries:

- **S:** Selects a report or source listing to be viewed.
- **C:** Accesses a utility for identifying a programmer with a job and how that person may be contacted. Available only for the Abend-AID Dataset Directory.
- **L:** Locks a report or source listing in the file. While locked, it cannot be automatically deleted when the file becomes full. The most current listing is automatically locked.
- **U:** Unlocks a previously locked report or source listing.
- **D:** Deletes a report or source listing from a file.
- **P:** Prints a report or source listing. Presents Hardcopy Options if Show print setup is active on the Entry Panel.

You can enter multiple line commands, which are executed from top to bottom.

Notes:

1. Depending on how the file was formatted, locking of source listings may not be permanent.
2. If you choose to continue without source support or continue with the most recent available source listing, your selection remains in effect until you exit the Abend-AID report or enter the SOURCE command.

Hardcopy Options

The Compuware/VF Hardcopy Options screen (Figure 13-5), shown with default settings, enables you to change specifications for printing reports and source listings. Obtain this screen by specifying **YES** at Show print setup on the Entry Panel and then entering the **P** (Print) line command in the report or source listing directory.

Figure 13-5. Compuware/VF Hardcopy Options Screen

```
 COMPUWARE/VF HARDCOPY OPTIONS
 COMMAND INPUT ===>

 Current DDIO file: AA.REPORTS
            Member: SBL60LN3
Report/List Number: 118

Enter/verify options below:
   Line size           ===> NARROW               (NARROW or WIDE)
   Page size           ===> 60                   (20 to 99)
   Sysout class        ===>                      (class,writer-name,format)
   Destination         ===>                      (nodename,userid)

   Route               ===> SYSOUT               (SYSOUT or BATCH)
   Max number of pages ===>                      (for Route=BATCH)
   Unit                ===> SYSDA                (for Route=BATCH)
```

Make the following entries on the Compuware/VF Hardcopy Options screen:

CURRENT DDIO FILE

The file name from which you are printing a member.

MEMBER

The member you selected to print.

REPORT/LIST NUMBER:

The report or list number of the member you selected to print.

Line size

Specify the width of the printed report or source listing.

NARROW	Default. Prints optional 80-column output.
WIDE	Prints standard 133-column output.

Page size

Enter the number of lines per page for printed output.

Sysout class

Specify the JES spool class where the output is printed.

Destination

Enter the JES destination code where the output is routed.

Route

Specify how the report or source listing is to be printed. Enter one of the following:

SYSOUT	TSO builds a SYSOUT file. Job information is obtained from the security profile attached to your TSO account. Your user ID is used as the JOBNAME.
BATCH	TSO allocates a work file into which the member is copied. A print job is submitted to JES using the default job card from your ISPF0.2 screen.

Max number of pages

Determine the size of the work file when the BATCH route value is specified.

Unit

Enter the device type for the work file when the BATCH route value is specified. It can be any valid physical address, device type, generic name, or VIO. (For example, use 2BF, 3380, SYSDA, or VIO).

Output Selection Menu

After you select an Abend-AID report for viewing, the Output Selection Menu is displayed. Figure 13-6 shows the Output Selection Menu.

Figure 13-6. Output Selection Menu

```
COMPUWARE/VF ---------------- OUTPUT SELECTION MENU ----------------------------
SELECT SECTION ===>

'AA.REPORT'        REPT.(L) 66  FLOCON01  J25586

   0 or HEADER   - System, job, and client information
   1 or DIAGS    - Diagnostic information
   2 or NSI      - Next Sequential Instruction (location in program)
   3 or TRACE    - Trace of program flow and program attributes
   5 or REGS     - Register contents and PSWs
   6 or PROG     - Program Storage
   7 or PLIST    - Program Listing (XLS only)
   8 or FILES    - Data management control blocks
   9 or IMS      - IMS diagnostic information and PCBs
  10 or IDMS     - CA-IDMS diagnostic and status information
  11 or DB2      - DB2 diagnostic and status information
  12 or SORT     - Current COBOL sort information
  13 or DATALOC  - COBOL Data Locator
  14 or MQ       - WebSphere MQ diagnostic information
  15 or LE       - LE information
  16 or EPILOG   - Abend-AID termination information

               Key section number or name and press ENTER.
   Or press ENTER to start at top of output and view sections 1 through 5.
```

The menu lists all report sections. All sections will not be applicable to all faults. If you select an inapplicable section a SECTION NOT FOUND message is displayed.

In the SELECT SECTION field at the top of the screen, enter the section number or the abbreviation for the section name.

If you select any section number from 0 through 5, all existing sections with numbers 0 through 5 can be viewed. If you select any section number greater than 5, only the selected section can be viewed. You can use standard ISPF scroll commands to scroll within the viewable portion of the Abend-AID report.

To view section 0 through section 5, press the Enter key. The Error Analysis section will be displayed. Use scrolling to view other sections. The sections provide the following information:

0 or Header

Top of output, including the header information and available help information.

1 or DIAGS

Error Analysis information.

2 or NSI

Error Location section.

3 or TRACE

Call Trace Summary and Application Program Attributes.

5 or REGS

Register contents and PSW.

6 or PROG:

Program storage for each program. If more than one program is present, a selection list is displayed. Enter an **S** next to the program storage to be viewed.

7 or PLIST:

XLS only. Displays the Procedure Divisions for COBOL programs on the calling chain and source listings for PL/I and Assembler programs on the calling chain with the current statement indicated.

8 or FILES:

File and current data management control block information. If more than one file is present, a selection list is displayed. Enter one of the commands shown to access a file through Compuware/VF or through File-AID. To display a specific file, enter **FILE(ddname)** in the COMMAND INPUT field.

9 or IMS:

Abend-AID for IMS information.

10 or IDMS:

Abend-AID for IDMS information.

11 or DB2:

Abend-AID for DB2 information.

12 or SORT:

Current sort record for internal COBOL sorts.

13 or DATALOC:

Identifies data strings in COBOL programs.

14 or Abend-AID for WebSphere MQ:

Displays either MQSeries batch or MQSeries IMS information created by Abend-AID for WebSphere MQ.

15 or LE:

Displays the following LE information:

- LE run-time options at the time of the error
- Indicates how the option was set and from where it was set
- Provides a complete validation and analysis of LE heap storage.

16 or EPILOG:

Abend-AID termination information.

Source Warnings

You receive one of the notification screens shown below if Abend-AID XLS can't find a suitable source listing for the report and the site is licensed for XLS.

You activate the notification when you try to view one of the report sections that present source information. All except Figure 13-10 on page 13-12 can also be obtained with the SOURCE MISMATCH command. The warnings are specific to the language being diagnosed, as noted.

Explanations of the screen choices follow the figures.

Figure 13-7. COBOL Mismatched Date and Time Warning

```
COMPUWARE/VF ------------------------------------------------------------------
COMMAND INPUT ===>

'PFHKXLO.CSS79.QA.REPTFILE'  REPT.4  EFHRIP0C  J31098

     * * * * * * * * * * * *    W A R N I N G    * * * * * * * * * * * *
     *       Program                            Most recent listing      *
     *       ------------------------------------------------------------ *
     * Name: COBSAMP                            COBSAMP                  *
     * Load: COBSAMP                                                     *
     * Date: 07 DEC 2000                        27 DEC 2001              *
     * Time: 11:22:20                           08:07:38                 *
     *                                                                   *
     * The program date and time stamp does not match a source listing.*
     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

 Type an S next to the desired option and press ENTER.

 _ Use the MOST RECENT source listing shown above.
 _ Display available source listings.
 _ Retrieve a source listing by submitting background JCL.
 _ Provide BASIC support without a source listing.

 Enter END command to exit.
```

Figure 13-8. PL/I Mismatched Source Listing Warning

```
COMPUWARE/VF ------------------------------------------------------------------
 COMMAND INPUT ===>

 'PFHKXLO.CSS79.QA.REPTFILE'  REPT.1  EFHRIP0C  J20788

      * * * * * * * * * * * *    W A R N I N G    * * * * * * * * * * * *
      *                                                                   *
      *         An appropriate source listing could not be found.         *
      *         Either:                                                   *
      *           1. a source listing for program PLIA5    does not       *
      *              exist, or                                            *
      *           2. a source listing exists, but it does not match       *
      *              the compiler language and release of the program     *
      *              in the Abend-AID report.                             *
      *                                                                   *
      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *


    Type an S next to the desired option and press ENTER.
        _ Retrieve a source listing by submitting background JCL.
        _ Provide BASIC support without a source listing.


  Enter END command to exit.
```

Figure 13-9. Assembler Mismatch Warning

```
COMPUWARE/VF ------------------------------------------------  ROW 1 TO 28 OF 31
 COMMAND INPUT ===>                                            SCROLL ===> CSR

     CSECT Name: EUR001
  Wildcard Mask: ********
      Assembled: 16 AUG 2000            Load module: EUR001

    The CSECT name and date do not uniquely match a source listing.

  _ Retrieve a source listing by submitting background JCL.
  _ Provide BASIC support without a source listing.

    Type an S next to the correct source listing below and press the ENTER key.
                ASSEMBLER SOURCE LISTING DATASET DIRECTORY
    PROGRAM    LIST NUMBER     ASMB DATE     TIME    RC   LANGUAGE    SIZE(K)
   ---------- 'EFHAXSO.TEST.DDIO' ----------------------------------------
   X80102B   (L)      3537    09 NOV 2001  12.20     00   HLASM R4     399
   TEMPNAME  (L)      3344    10 SEP 2001  13.33     00   HLASM R4     160
   TEST1     (L)      3179    22 JUN 2001  11.35     00   HLASM R4     160
   VIBAIITF  (L)      3046    28 MAR 2001  13.07     00   HLASM R4     559
   CWDEMASM  (L)      2612    14 JUN 2000  20.08     04   HLASM R3     240
   ASM03T    (L)      2651    08 JUN 2000  15.47     00   HLASM R3      80
   ASMPRG1   (L)      2330    13 MAR 2000  12.14     00   HLASM R3      80
   AIBHLP00  (L)      1023    T PAGO N     10.51     00   HLASM R2     719
```

Figure 13-10. Source-Not-Found Warning

```
COMPUWARE/VF ------------------------------------------------------------------
 COMMAND INPUT ===>

 'EFHRIP0.RPTFILE'  REPT.92  EFHRIP0C  J20788

      * * * * * * * * * * * *   W A R N I N G   * * * * * * * * * * * *
      *                                                                 *
      *         An appropriate source listing could not be found.       *
      *         Either:                                                 *
      *           1. a source listing for program COMP5    does not     *
      *              exist, or                                          *
      *           2. a source listing exists, but it does not match     *
      *              the compiler language and release of the program   *
      *              in the Abend-AID report.                           *
      *                                                                 *
      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *


    Type an S next to the desired option and press ENTER.
        _ Retrieve a source listing by submitting background JCL
        _ Provide BASIC support without a source listing
```

From these screens, choose as applicable:

Use the most recent source listing shown above

Abend-AID XLS will use the source listing that has a time stamp closest to that of the report.

Display available source listings

Abend-AID XLS presents a source listing directory, similar to Figure 13-9, that shows all members in the source datasets specified on the Entry Panel. All directory commands are valid.

Provide basic support without a source listing

Opens the specified report section, which will contain no source information.

Retrieve a source listing by submitting background JCL

Presents a dialog box that facilitatesthe retrieval of the source listing using the Language Processor.

The END command presents the Output Selection menu.

Chapter 14.
Printing & Utilities

Change Printing Options

Print a report or source listing by entering the P (Print) line command in a Compuware/VF dataset directory. You cannot print from within a report or source listing.

Change printing specifications on the Hardcopy Options screen. Access Hardcopy Options by having Yes specified at the Show print setup field of the Compuware/VF Entry Panel when you enter the Print command.

1. At the Compuware/VF Entry Panel, in the Show print setup field, specify **YES**.
2. On a Dataset Directory screen, enter **P** next to the job that you want to print.
3. At the resulting Hardcopy Options screen, change or check the printing specifications as necessary. Press Enter to print. New specifications will remain in effect until changed.

Figure 14-1. Compuware/VF Hardcopy Options Screen

```
 COMPUWARE/VF HARDCOPY OPTIONS
 COMMAND INPUT ===>

 Current DDIO file: AA.REPORTS
            Member: SBL60LN3
Report/List Number: 118

Enter/verify options below:
   Line size           ===> NARROW               (NARROW or WIDE)
   Page size           ===> 60                   (20 to 99)
   Sysout class        ===>                      (class,writer-name,format)
   Destination         ===>                      (nodename,userid)

   Route               ===> SYSOUT               (SYSOUT or BATCH)
   Max number of pages ===>                      (for Route=BATCH)
   Unit                ===> SYSDA                (for Route=BATCH)
```

4. You may change or leave as is the specification on the Show print setup field of the Compuware/VF Entry Panel. Continue as usual.

Utilities

Abend-AID XLS's utilities provide both support and operational capabilities.

Support utilities consist of installation and servicing functions. If you need information about those functions, refer to the utilities chapter of the *Abend-AID Installation and Customization Guide*. The operational utilities are used for report dataset and source listing dataset management as well as to prepare or change the processing for extended language support.

These are the operational utilities:

- The online file management commands of Abend-AID XLS's Compuware/VF are available in the Compuware/VF dataset directories, which are explained in this manual.
- The online Compuware Language Processor management functions of the CSS Utilities enable applications or systems programmers to quickly prepare or modify a Compuware Language Processor operation without manually changing JCL. Access the CSS Utilities on the Abend-AID XLS Primary Menu.
- The batch file management commands of Compuware Shared Services's CWDDSUTL utility provides complete management functions, including a file comparison program.

Refer to the *Compuware Shared Services User/Reference Guide* if you need information about the CSS Utilities or the batch file management utility.

Appendix A.
Testing Samples

This table lists sample programs (IVPs) containing faults. The programs may be used in Abend-AID testing and training. They are kept in the Abend-AID installation library (SORCMAC).

Note: Any permanent datasets created by Abend-AID XLS IVPs are allocated using the installer's TSO ID as the high-level qualifier.

In some member names provided below, v represents the DB2 version number and xxxx is the DB2 subsystem ID.

Table A-1. Testing Samples

Program name	Tests	Language	Fault/Purpose
$71ABTST	Base report	Assembler	B37, D37, 0C7, 0C4
$74ABSPF	Compuware/VF	Assembler	B37, D37, 0C7, 0C4
$72LECOB	Compuware/VF	COBOL LE	0C7
$73LEPLI	Compuware/VF	PL/I LE	Oncode 0320
$75XLSC1	XLS report	COBOL	0C7
$75XLSC2	XLS report	COBOL	0C7
$75XLSC3	XLS report	COBOL	0C7
$75XLSC4	XLS report	COBOL	0C7
$75XLSP1	XLS report	PL/I	Oncode 0320
$75XLSP2	XLS report	PL/I	Oncode 0320
$75XLSP3	XLS report	PL/I	Oncode 0320
$75XLSPA	XLS report	PL/I	U3020
$75XLSPB	XLS report	PL/I	U3020
$75XLSA1	XLS report	Assembler	0C7
$75XLSA2	XLS report	Assembler	0CB
$75XLSA3	XLS report	Assembler	U0200
$75XLSA4	XLS report	Assembler	0C7
$81vxxxx	DB2	COBOL	DB2, Plan, -302 or -310
$82vxxxx	DB2	COBOL	DB2, Package, -302 or -310
$83vxxxx	DB2	COBOL	DB2 V4, Stored Procedures, -407
$84vxxxx	DB2	COBOL	DB2 V5, Stored Procedures, -407
$85vxxxx	DB2	COBOL	DB2 V6, Stored Procedures, -407
$86vxxxx	DB2	COBOL	DB2 V7, Stored Procedures, -407
$91vxxxx	DB2	PL/I	DB2, Plan, -302 or -310
$92vxxxx	DB2	PL/I	DB2, Package, -302 or -310
$93vxxxx	DB2	PL/I	DB2 V4, Stored Procedures, -407
$94vxxxx	DB2	PL/I	DB2 V5, Stored Procedures, -407
$95vxxxx	DB2	PL/I	DB2 V6, Stored Procedures, -407
$96vxxxx	DB2	PL/I	DB2 V7, Stored Procedures, -407
$52FORTS	DB2	FORTRAN	FORTRAN, FORTRAN in LE
$42PLI02	DB2	PL/I	U3020

Glossary

abend. ABnormal END of task. The termination of a job, prior to normal completion, caused by an unresolved error condition.

abend code. A three-digit, hexadecimal system completion code or a four-digit, decimal user completion code for an abnormal end.

ACB. VSAM access method control block.

ATE. Language Environment's Abnormal Termination Exit. LEAID is the Abend-AID XLS provided exit.

Abnormal Termination Exit (ATE). Language Environment's Abnormal Termination Exit. LEAID is the Abend-AID XLS provided exit.

basic direct access method (BDAM). File access method that directly retrieves or updates specified blocks of data on a direct access storage device.

basic language support. Abend-AID reports without source information. Compare to extended language support. Release 9.4 and later of Abend-AID require extended language support.

batch. Processing in which jobs are grouped (batched). The jobs are executed sequentially, and each job must be processed to completion before the following job can begin execution.

batch file utility (CWDDSUTL). A utility used for the preparation of report and source listing files.

BDAM. Basic direct access method.

BL cell. Base locator cell. Contains the base address used by the computer to reference data within storage. These cells are used by COBOL programs to provide addressability to data within the working storage section of a program.

BLL cell. Base linkage locator cell. These cells are used by COBOL programs to provide addressability to data within the linkage section of a program.

CAA. Language Environment's Common Anchor Area.

CA-IDMS. Computer Associates' Integrated Data Management System that runs on IBM computers and provides a natural language query language.

CEEDOPT. Language Environment's default batch option control table.

CEEROPT. Language Environment's option control table for IMS regions.

CEEUOPT. Language Environment's user option control table.

CIB. Language Environment's Condition Information Block.

CLIST. Condensed listing of the Procedure Division for a COBOL program.

COBOL. Common business-oriented language. A high-level programming language, based on English, that is used primarily for business applications.

Compuware language processor. A component that accepts COBOL, PL/I, and Assembler compiler output, builds sort work records, sorts and merges the records, and merges the records with the output listings to produce processor control blocks for use as input to Abend-AID XLS.

Compuware Shared Services (CSS). Storage, retrieval, and maintenance capabilities provided for reports and source listings by components shared among mainframe Compuware products.

Compuware/VF (Compuware/VF). A menu-driven, online viewing facility used by Abend-AID XLS.

Compuware/VF. Compuware Viewing Facility.

CSECT. Control section.

CSS. Compuware Shared Services

CWDDSUTL. The main program for the batch file utility and Compuware/VF.

data name. The name of a data item in the Data Division of a COBOL program.

DB2. DATABASE 2. IBM's database management system that provides a relational model of data. DB2 runs as a subsystem of MVS.

distributed viewing. A means of accessing Abend-AID XLS report and source listing files on remote MVS systems that don't share DASD.

DL/I. Data language one. IBM's database management facility provided by the IMS/VS database program products.

DMAP. Data Division map for a COBOL program.

DSECT. Dummy control section. A control section that an assembler can use to format an area of storage without producing any object code.

dump. Hexadecimal representation of storage that may contain data useful for diagnosing an error.

EDB. Language Environment's Enclave Data Block.

enhanced compiler listing. A convenient source of quick reference information and program documentation that merges MAP/DMAP and OFFSET/CLIST information, in addition to error and diagnostic messages, with a source listing.

Enterprise Common Components (ECC). The shared components of mainframe Compuware products, that provide Compuware Shared Services.

entry-sequenced dataset (ESDS). VSAM dataset whose records are loaded in sequence. Unlike a normal sequential dataset, ESDS records can be accessed randomly by their addresses.

ESDS. Entry-sequenced dataset.

extended language support (XLS). An Abend-AID XLS facility that utilizes a language processor and Compuware/VF components to provide source information in the Abend-AID XLS report.

Heap Segment. LE heap storage is comprised of one or more segments. A heap segment contains both allocated and free storage elements.

Heap Storage. An internal storage structure used by LE storage management services.

HELP. Primary command that requests Abend-AID XLS's interactive Help facility.

HLASM. High-level Assembler

IAM. Innovation Access Method

IDMS. See *CA-IDMS*.

IMS. Information Management System. IBM's DB and TM database systems capable of managing complex databases and networks.

ISAM. Indexed sequential access method.

JCB. Journal control block.

JCL. Job control language.

key. Code used to locate a record and establish its position in an index. The key can be part of a field, a full field, or multiple fields duplicated from the record.

key-sequenced dataset (KSDS). VSAM file type whose records are loaded in key sequence. Records are retrieved by key or address using an index. New records are inserted in key sequence by means of distributed free space.

KSDS. Key-sequenced dataset.

Language Environment. The run-time environment of IBM's z/OS operating system.

LEAID. The ATE provided by Abend-AID XLS for Language Environment error handling.

linkage section. A section of a COBOL program used to describe data that is passed to it from MVS or another program.

MCH. Language Environment's Machine State Information Block.

Multiple Virtual Storage (MVS). Actual name is OS/VS2-MVS. Operating system for large IBM mainframe computers.

MVS. Multiple virtual storage.

OCB. Language Environment's runtime Option Control Block.

offset. A relative location or position within a data area.

operating system. Software that controls the execution of jobs. It may provide resource allocation and scheduling.

paragraph. A set of one or more COBOL sentences, making a logical processing entity, and preceded by a paragraph name or a paragraph header.

PCB. Language Environment's Process Control Block.

PCB. DL/I Program Communication Block.

PL/I. Programming language one. A programming language designed for numeric scientific computations, business data processing, systems programming, and other applications.

Procedure Division. The section of a COBOL program that contains executable instructions.

program communication block (PCB). A control block used by DL/I to define the databases that can be accessed by a particular PSB.

program specification block (PSB). DL/I control block that defines a set of DL/I databases that can be accessed from a program.

program status word (PSW). An operating system control block defining the current status and location of a program that is executing.

program storage. A class of storage used for application programs.

PSB. Program specification block.

PSW. Program status word.

register save area. A group of 72 contiguous bytes used for saving registers when one program calls another.

relative record dataset (RRDS). VSAM dataset whose record locations are specified by a number that represents a record's location in the dataset relative to the beginning of the dataset.

report dataset. A dataset used to route report output and referenced when viewing reports using Compuware/VF. Specifically, report datasets are report shared directories, report databases attached to a shared directory, and "DDIO" report files.

RRDS. Relative record dataset.

RRSAF. DB2 Recoverable Resource Manager Services Attachment Facility.

RTLS. LE Runtime Library Services.

search segment argument (SSA). A control block used by DL/I to access a segment within the hierarchy of a database.

SNAP-AID. Abend-AID XLS facility that displays Abend-AID XLS output during program execution without forcing the program to terminate.

snap dump. A dump that is taken at a specific point during execution of a program. Processing is generally continued after the dump has been taken.

source listing. A compiler listing and other information about a program stored in a source listing dataset and accessed by Abend-AID XLS.

source listing dataset. A dataset used to store source listings and referenced when viewing reports using Compuware/VF. Specifically, source listing datasets are source listing shared directories, source listing databases attached to a shared directory, and "DDIO" source listing files.

SQL. Structured query language.

SSA. Search segment argument.

statement numbers. Sequence numbers provided by compilers and assemblers to give the programmer an easy means of identifying statements within a program.

TGT. A COBOL control block that is also known as the memory map. Contains information about COBOL internal operations and essential information for COBOL debugging.

U4039-8. The abend code issued by the Abend-AID XLS ATE (LEAID) when a system dump is requested in addition to the Abend-AID XLS report.

virtual storage access method (VSAM). File access method whereby the records in a file on a direct access storage device can be accessed in key sequence (KSDS), entry sequence (ESDS), or relative record sequence (RRDS).

VSAM. Virtual storage access method.

Working-Storage Section. A section of a COBOL program used to define the data items that are used in a program.

XLS. Extended Language Support.

Index

A

B

C

D

E

F

G

H

I

J

L

M

O

P

R

S

T

U

V

W